RL-TR-97-202, Volume II (of two)
Final Technical Report
October 1997

MOLECULAR DYNAMICS SIMULATION UPGRADE - SOFTWARE USER'S MANUAL

Synectics Corporation

Lisa Kolek and Geraldine W. Rogers

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-97-202, Volume II has been reviewed and is approved for publication.

APPROVED: Herbert F. Helbig
HERBERT F. HELBIG
Project Engineer

FOR THE DIRECTOR: John J. Bart
JOHN J. BART
Acting Director, Reliability
Electromagnetics & Reliability Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE October 1997	3. REPORT TYPE AND DATES COVERED Final Jan 97 - Jun 97

4. TITLE AND SUBTITLE

MOLECULAR DYNAMICS SIMULATION UPGRADE - SOFTWARE USER'S MANUAL

5. FUNDING NUMBERS

C - F30602-95-D-0028/0009
PE - 61102F
OR - 2982
TA - QE
WU - 10

6. AUTHOR(S)

Lisa Kolek and Geraldine W. Rogers

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Synectics Corporation
111 East Chestnut Street
Rome NY 13440-2831

8. PERFORMING ORGANIZATION REPORT NUMBER

WH-94-RY-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratory/ERDR
525 Brooks Rd
Rome NY 1344-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-97-202, Volume II (of two)

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Herbert F. Helbig/ERDR/(315) 330-3495

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of the MDEM simulation upgrade was to improve performance, enhance the "user-friendly" features and generally clean up the source code for the MDEM simulation software tool. Performance improvement was primarily focused on the attainment of MDEM processing speedup. This was achieved at a low level by examining the existing code structure, as well as at a high level by modifying the simulation algorithm. Enhancement of user-friendly features included the implementation of a prototype graphical user interface (GUI) and the creation of the user's guide for both the command-line and GUI versions of MDEM. Code cleanup involved deletion of unnecessary code and comments, and reorganization of variables. In addition to this, miscellaneous software engineering and development support was provided to Rome Laboratory on an "as required" basis.

14. SUBJECT TERMS

Molecular Dynamics, Computer Simulation, Electronic Materials

15. NUMBER OF PAGES

68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

List of Exhibits

List of Tables

SECTION ONE
GENERAL OVERVIEW

This Software User's Manual (SUM) presents the information and instructions necessary for user interaction with the Molecular Dynamics Simulation of Electronic Materials program, "MDEM." The information contained herein will instruct the user about the procedures required to install and carry out the operation of the MDEM software.

This report satisfies the Contract Data Requirements List (CDRL) A003 according to DI–MISC–80711/T, Scientific and Technical Reports, which specifies the document's format be in accordance with ANSI Z39.18–1987, "Scientific and Technical Reports: Organization, Preparation, and Production."

1.1 APPLICATION OVERVIEW

Atomic motion reduces the useful life of microcircuits. Rome Laboratory's (Rome Lab's) approach was to develop simulation tools to investigate the three–dimensional atomic motion and structure. This solution uses molecular dynamics simulation to examine the growth and transport properties of polycrystalline structures at the molecular scale. The purpose of MDEM, therefore, is to reveal the atomic–level mechanisms that are responsible for the degradation of microelectronics components and interconnects.

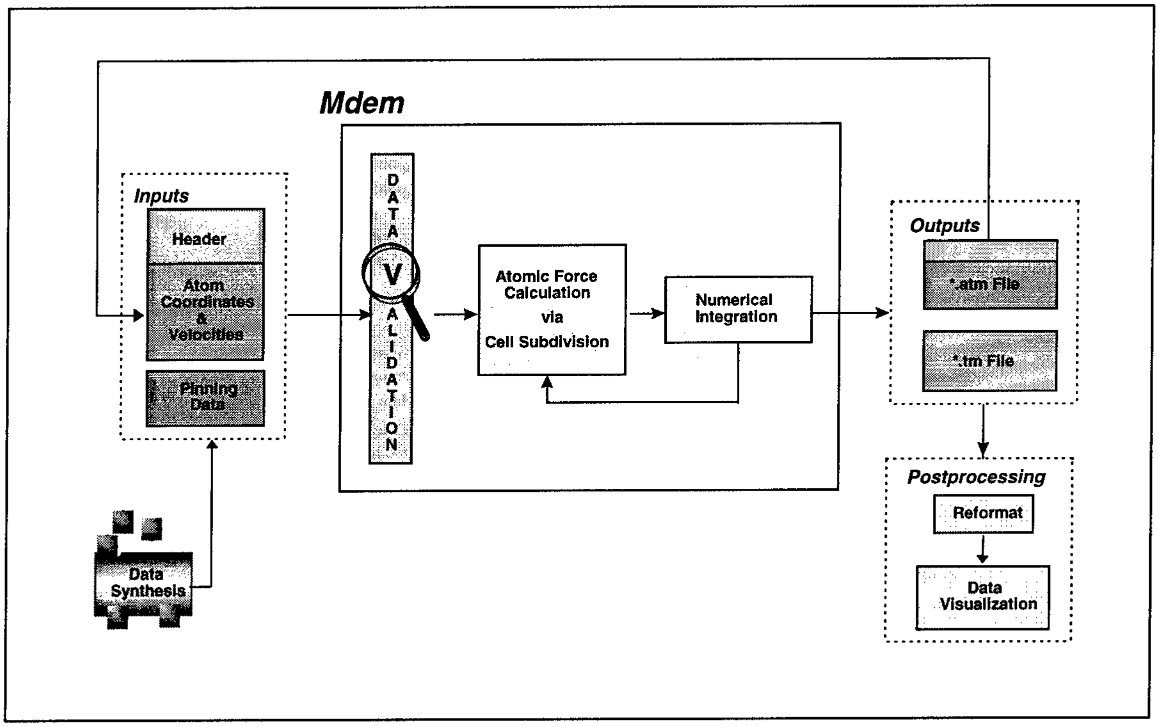
The MDEM software reads in a user–supplied data set consisting of such input parameters as initial system temperature, number of integration steps, cutoff distance, x, y, and z coordinates and velocity components for the atoms, and a tag denoting each atom's species, thermal, and grain properties. The user can supply an optional data set which describes the points to which atoms may be pinned by elastic forces. Data validation functions check for the presence of data, as well as checking to see that the values are within valid ranges. Next, MDEM performs the numerical integration of f=ma according to Stanley's STA[1]. To efficiently perform the interatomic force calculations required by the integration, a grid of cells is overlaid on the atom data. This method of partitioning the data results in non–redundant pairing of atoms by examining the interaction between each cell and half of its adjacent neighbors. The data are written to file periodically, at a frequency set by the user in either the input file or from MDEM's graphical user interface (GUI).

[1] Robert W. Stanley. "Numerical Methods in Calculation". American Journal of Physics, volume 52, number 6, p. 499 (June 1984)

The main output of the simulation is a data set, listing the atoms' new coordinates, velocities, and the atoms' tags. A secondary output file, optionally chosen by the user, is the thermostat file. The user can plot the thermostat data to determine changes in temperature.

With molecular visualization packages, such as *RasMol,* which render molecular coordinates on the screen in a three–dimensional representation, the reformatted output from MDEM can be examined for density fluctuations due to shock or sound waves, reconstruction of a surface to achieve a lower energy configuration, and abrupt changes in regularity between regions of well–formed crystallinity. Exhibit 1 illustrates the functionality of MDEM in an end–to–end molecular dynamics simulation.

Exhibit 1. Molecular Dynamics Simulation Using MDEM



MDEM is a Rome Lab program, written and conceived by Rome Lab engineers and scientists. Synectics assisted Rome Lab in algorithm analysis and implementation, code cleanup, GUI design and implementation, and in preparing this SUM.

1.2 OPERATIONAL ENVIRONMENT

MDEM has two operational versions — command–line (CL) and graphical user interface (GUI). For both versions, the assumption is that MDEM resides locally. In its current format it is not intended for distributed processing.

The CL version is designed and coded to maximize the portability and flexibility of today's high performance personal computers and workstations. It is written in ANSI C.

The GUI version is written for the 32–bit personal computer (PC). It was developed using Microsoft's Visual C++, with calls to the Microsoft Foundation Classes (MFC).

The memory requirements for the executable code for both the CL and the GUI versions are under 300 KB. The amount of RAM and swap space depends upon the number of atoms in the data set.

1.3 USER INTERACTION

MDEM is designed to be an interactive system. However, while the CL version requires interaction, scripts can be written for automated operation of MDEM. The GUI version of MDEM uses the Windows Graphical User Interface to provide its look and feel. Use of a PC running a 32–bit operating system is required for the GUI version. For users unfamiliar with file transfer operations on the most common 32–bit PC operating system, "Windows 95," a brief primer has been included in Section 3.1.

This manual introduces the user to the complete functionality of MDEM. It teaches the user how to implement these functions. The anticipated audience for the MDEM software are those familiar with molecular dynamics simulation. This manual, therefore, does not discuss how to analyze the results of the data.

1.4 CONVENTIONS USED

Certain conventions are used in this MDEM's SUM to help the user understand the commands more easily. Table 1 describes these conventions.

Table 1. Conventions

	DESCRIPTION	EXAMPLE
`special font`	This format is used to represent the MS–DOS command–line prompt.	`C:\MDEM\>`
`bold special font`	This format shows user input, such as commands, command options, and names of directories and files used as arguments.	**`mdem4`** *`06141555.25`*
`italic special font`	This font shows the names of variable elements to which the values are given by the user, such as a filename or response.	*`06141555.25`*
SMALL CAPS BOLD FONT	This font indicates the name of a dialog, such as **VIEW ATOMS** pop–up dialog, or the name of a data entry field.	1. Enter a number in the **TM FILE WRITE INTERVAL** field.
type>	This style indicates that the information following the > must be entered via the keyboard.	1. type>
Run MDEM	This symbol indicates a special button on the **MDEM** dialog.	Select the Run MDEM button
☞	This symbol precedes a special note to the user. The note is offset by a box.	☞ The Run MDEM button is disabled during processing
1.	This numbered step indicates that an action to be taken by the user, such as type or select, follows.	1. type> **`mdem4`** ***`06141555.25`***
♦	This symbol indicates a second or alternative action to be taken by the user.	♦ Or, use the “drag–and–drop” method to drop the input file *`06141555.25`* onto MDEM4.exe
⇒	This symbol indicates a description of the last action.	⇒ displays the data file’s input parameters
<CR>	This notation indicates that the user should press the “Return” or “Enter” key on the keyboard.	1. type> **`mdem4`** ***`06141555.25`*** <CR>

	DESCRIPTION	EXAMPLE
<CTRL>	This notation indicates that the user should press the "Control" key on the keyboard, usually in conjunction with another key.	☞ Pressing <CTRL><C> will cancel the integration.

The following instruction combines the conventions described in Table 1.

1. type> **mdem4** *data*

   ⇒ where **mdem4** is the name of the MDEM executable and *data* is the input data file name

   ⇒ displays the data file's header information (list of parameters for the run)

   ♦ Or, use the "drag–and–drop" method to drop the input file onto MDEM4.exe

1.5 REFERENCED DOCUMENTS

Government and non–Government documents used during the preparation of this document are referenced below.

1.5.1 GOVERNMENT DOCUMENTS

□ DI–IPSC–81445 (Software Input/Output Manual — SIOM)

□ DI–IPSC–81444 (Software Center Operator Manual — SCOM)

□ DI–MCCR–80019A (Software User's Manual — SUM)

1.5.2 NON–GOVERNMENT DOCUMENTS

□ "The Art of Molecular Dynamics Simulation," D.C. Rapaport, Cambridge University Press, New York, NY, 1995.

□ DJGPP Web Site, http://www.delorie.com/djgpp

❑ Microsoft Visual C++ Version 5.0 on–line documentation

❑ *"RasMol* v2.5: A Molecular Visualization Program User's Manual," Roger Sayle, Biomolecular Structure, Glaxo Research and Development, Greenford, Middlesex, UB6 0HE., U.K., October 1994.

1.6 ASSISTANCE AND PROBLEM REPORTING

Address MDEM inquiries to:

Herbert F. Helbig
Rome Laboratory/ERDR
525 Brooks Road
Rome, NY 13441–4505
e–mail: helbigh@rl.af.mil

1.7 DOCUMENT OVERVIEW

This manual is written for the most generic use of MDEM. It attempts to provide answers to questions which may arise, along with all information needed to use the MDEM software effectively.

Section 1 of this manual presents an overview of the MDEM application including its purpose, capabilities, operational environment, user interaction, the notational conventions used in this manual, and references.

Section 2 details the inputs required from the MDEM software as well as the expected outputs. Examples of each type of input and output are given.

Section 3 describes installation and operational procedures. It includes a brief primer on the Windows 95/NT file transfer techniques. Instructions are given for both the CL and the GUI versions of the MDEM software.

Section 4 contains information required for further development of the MDEM software, including a software inventory and instructions for setting up a project workspace for use with the Microsoft Visual C++ Development Studio.

Appendix A contains a list of commonly used acronyms.

Appendix B contains sample input files, main input file, and pinning point file.

Appendix C describes DJGPP, a freeware GNU C compiler, where to find DJGPP, and the flags used during compilation of the MDEM executable.

Appendix D contains notes on the various Microsoft Visual C++ code generation and optimization flags used during the development of MDEM.

Appendix E contains a brief description of *RasMol*, a molecular graphics viewer. Information is given on converting MDEM's *.atm file into a file format suitable for viewing the molecular structure using the *RasMol* software.

SECTION TWO
DATA REQUIREMENTS

This section describes the input data required by the MDEM software as well as the results. Included are examples of each, in the form of screen shots. It must be noted that Microsoft WordPad was the editor used to display the text files in the examples. By no means is the user limited to the use of WordPad. Any text editor can be used for editing either the input or output files.

2.1 INPUTS

Section 2.1 covers the rules and convention for input files for the MDEM software. Examples are given in the form of screen shots. An example of the entire input data set is located in Appendix B.

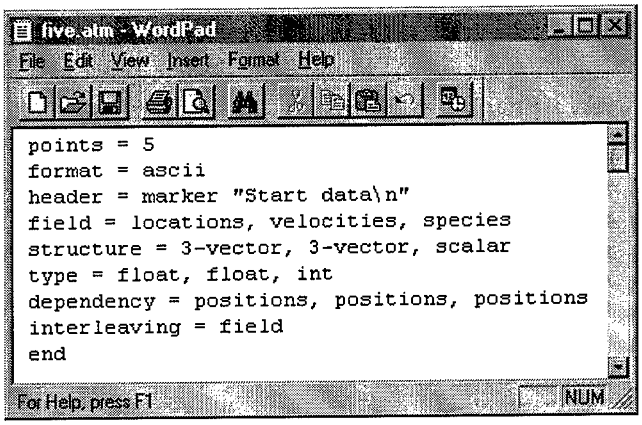
2.1.1 RULES AND CONVENTIONS

MDEM requires at least one input data set consisting of the system parameters and the atomic coordinates, velocities, and tag value. Furthermore, a separate data set consisting of an identification number and x, y, and z coordinates which serve as pinning points can be input. Currently, all inputs are in text format. The only naming convention restriction for the input file (sometimes referred to as *.atm) is that the own_name parameter in the input file header must match the file's name.

2.1.1.1 SYSTEM PARAMETERS

The system parameters, or header data, are one per line, with the input value(s) following the parameter name (see Table 2). There are three sections to the header data: file structure information, status data, and control data. The structure information section shown in this SUM is generated by IBM's "Data Explorer" software. It is not used by the MDEM software (Exhibit 2) except as part of the input data validation routines. Its presence facilitates viewing MDEM data with the "Data Explorer" software. Removal of this header will cause a data validation error ("Reached EOF too soon").

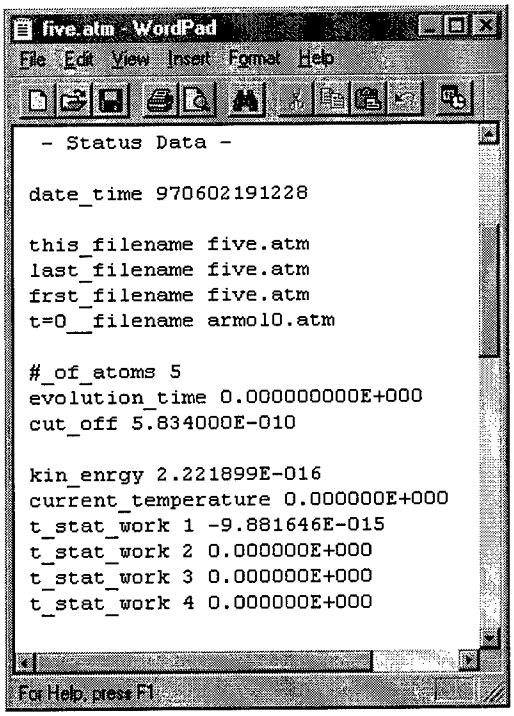
Exhibit 2. Data Explorer Header



```
points = 5
format = ascii
header = marker "Start data\n"
field = locations, velocities, species
structure = 3-vector, 3-vector, scalar
type = float, float, int
dependency = positions, positions, positions
interleaving = field
end
```

The status data section contains a record for the previous MDEM run (Exhibit 3). From status data the user can determine the evolution of the current data set. In addition, system values for the current run are set by the final values from the last run of MDEM.

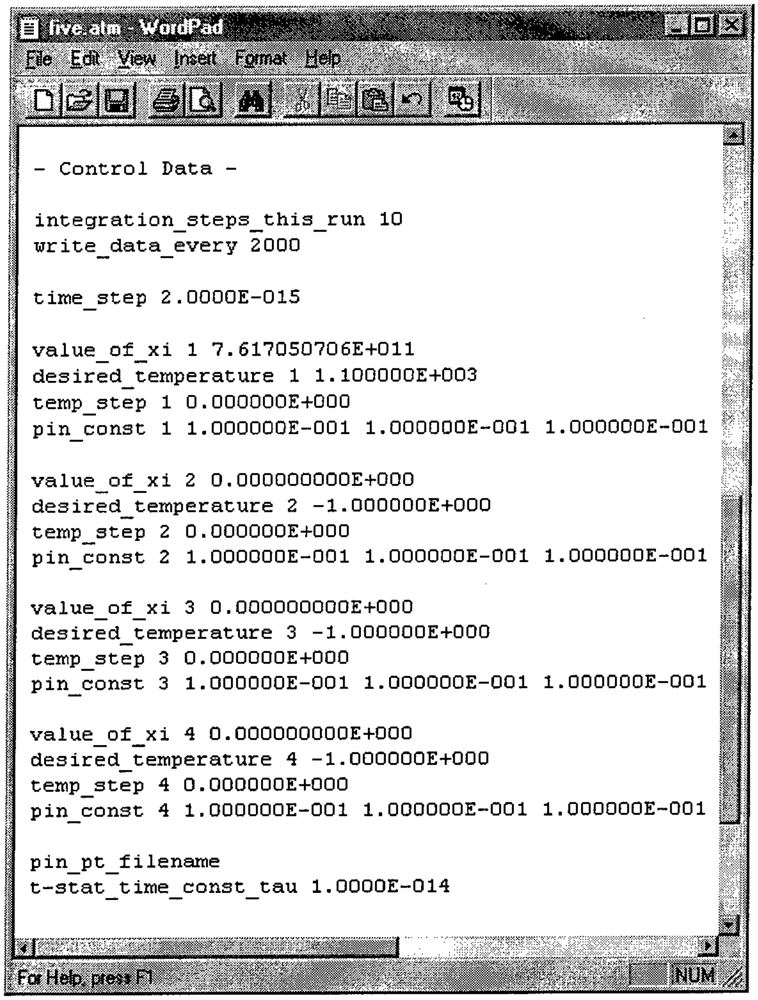
Exhibit 3. Status Data



```
 - Status Data -

date_time 970602191228

this_filename five.atm
last_filename five.atm
frst_filename five.atm
t=0__filename armo10.atm

#_of_atoms 5
evolution_time 0.000000000E+000
cut_off 5.834000E-010

kin_enrgy 2.221899E-016
current_temperature 0.000000E+000
t_stat_work 1 -9.881646E-015
t_stat_work 2 0.000000E+000
t_stat_work 3 0.000000E+000
t_stat_work 4 0.000000E+000
```

The control data section (Exhibit 4) consists of input values which bound the data run, such as the number of integration steps for the current run and how often to increase the system temperature.

Exhibit 4. Control Data



```
- Control Data -

integration_steps_this_run 10
write_data_every 2000

time_step 2.0000E-015

value_of_xi 1 7.617050706E+011
desired_temperature 1 1.100000E+003
temp_step 1 0.000000E+000
pin_const 1 1.000000E-001 1.000000E-001 1.000000E-001

value_of_xi 2 0.000000000E+000
desired_temperature 2 -1.000000E+000
temp_step 2 0.000000E+000
pin_const 2 1.000000E-001 1.000000E-001 1.000000E-001

value_of_xi 3 0.000000000E+000
desired_temperature 3 -1.000000E+000
temp_step 3 0.000000E+000
pin_const 3 1.000000E-001 1.000000E-001 1.000000E-001

value_of_xi 4 0.000000000E+000
desired_temperature 4 -1.000000E+000
temp_step 4 0.000000E+000
pin_const 4 1.000000E-001 1.000000E-001 1.000000E-001

pin_pt_filename
t-stat_time_const_tau 1.0000E-014
```

The input file requires a keyword identifier followed by the numerical value. These keywords are similar to, but not identical to, the source code's variables. Table 2 maps each keyword with the source code parameter; a one line description of each is also included.

Table 2. System Parameters

PARAMETER	KEYWORD	DESCRIPTION
Status Data		
hdr_date_time	date_time	creation date–time stamp for data set
own_name	this_filename	birth (file) name of data set
last_file	last_filename	data set from which this file was created

PARAMETER	KEYWORD	DESCRIPTION
Status Data (cont'd)		
frst_file	frst_filename	first file of run
eve_file	t=0_filename	record of earliest file (*.atm) in evolution
numatoms	#_of_atoms	number of atoms
elapsed_time	evolution_time	system evolution time of the last run in seconds
cut_off	cut_off	the force cutoff length in meters of the last run
ke	kin_enrgy	system kinetic energy of the last run in Joules
temp	current_temperature	system temperature in Kelvins
WTcum	t_stat_work	accumulated energy delivered by the thermostats per integration step in Joules
Control Data		
nsteps	integration_steps_this_run	number of integration steps to be done
write_intrvl	write_data_every	frequency of writing out data to file
del_t	time_step	integration time step in seconds
xi0	value_of_xi	initial thermostat function value in $(\text{seconds})^{-1}$
temp_step	temp_step	change in temp_ref per time step in Kelvins
temp_ref	desired_temperature	desired system temperature in Kelvins
pin_const	pin_const	array of pinning force constants in Newtons/meter
pin_file_name	pin_pt_filename	name of external pinning points file
tau	t_stat_time_const_tau	thermostat time constant in seconds

Most inputs have only one value. The exceptions are the pinning point constants (pin_const) and the thermostat related data items, including xi, t_stat_work, and temp_step. There can be a maximum of seven thermostat values used. For example, the data corresponding to thermostat 2 resides in array element 2 of all thermostat data arrays.

Table 3 presents the input "rules" — input length for character arrays, format conventions, sequencing of the data, punctuation, etc.

Table 3. Input Rules

PARAMETER	DATA TYPE	INPUT LENGTH	MIN. VALUE	MAX. VALUE	RESTRICTIONS	REQ'D
			Status Data			
hdr_date_time	char	[12]	n/a	n/a		✓
own_name	char	[13]	n/a	n/a		✓
last_file	char	[15]	n/a	n/a		✓
frst_file	char	[13]	n/a	n/a		✓
eve_file	char	[13]	n/a	n/a		✓
numatoms	ul[2]	✶[3]	1	none		✓
elapsed_time	double	✶	none	none		✓
cut_off	double	✶	none	none		✓
ke	double	✶	none	none		✓
temp	double	✶	none	none		✓
WTcum	double	[8]	none	none		✓
			Control Data			
nsteps	ul	✶	1	none		✓
write_intervl	ul	✶	0	none	must be >= 0	✓
del_t	double	✶	none	none	not equal 0	✓
xi0	double	[8]	none	none		✓
temp_step	double	[8]	none	none		✓
temp_ref[4]	double	[8]	none	none	not equal 0	✓
pin_const	double	[8]	none	none	not equal 0	
pin_file_name	char	[20]	n/a	n/a		✓[5]
tau	double	✶	none	none	not equal 0	✓

[2] ul — unsigned long integer

[3] ✶ — depends upon the target system's definition of the variable (integer, double, etc).

[4] temp_ref[] —holds the thermostat values. Atoms without thermostat tags will be assigned temp_ref[0] which should be negative. That forces their xi[0] to hold constant at xi0[]=0 so no thermostat force is applied in move_atoms().

[5] pin_file_name — identifies the name of the pinning point file optionally supplied by the user. The keyword is required but an associated file name is not. If a pinning point file name is not supplied, the system does not pin the atoms.

2.1.1.2 Atomic Data

The atomic data follows the "Start Data" indicator in the input file. The atomic coordinates, velocities, and tag values are in seven columns by number of atoms format. Two spaces separate each column, including the one column reserved for the negative sign. The atom tag is left–justified within its column.

The x, y, and z coordinates and velocities are read into MDEM as double precision floats. The atom tag is an integer value.

The atom tags are the encoded numbers representing information on each atom, such as species or atomic number, grain, pinning status, and thermostat number. A bitwise mask and shift of the atom tag is used to extract these properties. For example, when atom_tag[i] is masked with PIN_MASK and shifted by PIN_SHIFT, a value of one (1) indicates that the atom is pinned. A value of zero (0) indicates that it is not pinned.

The following masks and shifts, taken from "Mdemutl.h" are ANDed with atoms_tag[i] to extract information about atom i:

- ❑ SPECIES_MASK — isolates atomic numbers, integers up to 127, that are used to enter a lookup table of atomic masses which is in move_atoms() in mdemutl.c

- ❑ GRAIN_MASK — marks atoms according to their initial location so that visualization software can color them and reveal migrations. Eight possibilities are available by adding the various combinations of 0, 128, 256, 512 to atoms_tag[]. The combination 128 + 256 + 512 = 896 will tag an atom to be excluded from the system KE (and temperature) calculation. This is useful for deposition studies where the incoming atoms should not contribute to the substrate temperature until they become part of it. This interpretation is enforced in move_atoms. In Mdemutl.h, the GRAIN_MASK is defined as "896." GRAIN_SHIFT = 7.

- ❑ PIN_MASK — marks atoms that are pinned or immobilized in some way. Add 1,024 to atoms_tag[i] to pin atom i. In Mdemutl.h, the PIN_MASK is defined as "1024." PIN_SHIFT = 10.

- ❑ THERM_MASK — marks atom i as a thermostat set at one of 7 temperatures by adding combinations of 2,048, 4,096 and 8,192 to atoms_tag[i]. THERM_SHIFT is the number of bits of right shift that will make the masked data equal to 0, 1, 2, ..., 7. Mdemult.h defines the THERM_MASK as 14336 and the THERM_SHIFT as 11.

☞ Not all elements in the periodic table are supported by MDEM. Edit the arrays mass[ ] and rm_u0[ ] in mdemutl.c to add elements.

2.1.1.3 PINNING DATA

This section discusses the pinning point data file, an optional input to MDEM. If a pinning point file name is listed after the keyword "pin_file_name", MDEM attempts to locate and open this file. The pinning point file consists of four columns, each separated by two spaces. The first column contains integers that correspond to the atom's position in the main input file (i.e., the number 1 identifies the first atom in the main file, and so.). The values in the first column must be right–justified. The remaining three columns hold the x, y, and z coordinates used to pin atom[i]. These coordinates are double precision floats. The file can contain as many pinning points as desired as long as the number of atoms in the main input file are not exceeded. Exhibit 5 is an example of a pinning point file.

Exhibit 5. Pinning Point File

```
 1 -1.375325E-08 -6.978250E-09  0.000000E+00
 2 -1.375325E-08 -6.436250E-09  0.000000E+00
 3 -1.375325E-08 -5.894250E-09  0.000000E+00
 4 -1.375325E-08 -5.352250E-09  0.000000E+00
 5 -1.375325E-08 -4.810250E-09  0.000000E+00
 6 -1.375325E-08 -4.268250E-09  0.000000E+00
 7 -1.375325E-08 -3.726250E-09  0.000000E+00
 8 -1.375325E-08 -3.184250E-09  0.000000E+00
 9 -1.375325E-08 -2.642250E-09  0.000000E+00
10 -1.375325E-08 -2.100250E-09  0.000000E+00
11 -1.375325E-08 -1.558250E-09  0.000000E+00
12 -1.375325E-08 -1.016250E-09  0.000000E+00
13 -1.375325E-08 -4.742500E-10  0.000000E+00
14 -1.375325E-08  6.774978E-11  0.000000E+00
15 -1.375325E-08  6.097496E-10  0.000000E+00
16 -1.375325E-08  1.151750E-09  0.000000E+00
17 -1.375325E-08  1.693750E-09  0.000000E+00
18 -1.375325E-08  2.235750E-09  0.000000E+00
19 -1.375325E-08  2.777750E-09  0.000000E+00
20 -1.375325E-08  3.319750E-09  0.000000E+00
21 -1.375325E-08  3.861750E-09  0.000000E+00
22 -1.375325E-08  4.403750E-09  0.000000E+00
23 -1.375325E-08  4.945750E-09  0.000000E+00
24 -1.375325E-08  5.487750E-09  0.000000E+00
25 -1.375325E-08  6.029750E-09  0.000000E+00
```

☞ The above pinning point file is a valid example of a pinning point file. However, it does not correspond to the main input file shown as an example.

2.2 OUTPUTS

MDEM produces two types of output — information written to file and information displayed to the screen. MDEM writes a main output file automatically; the user must specifically request the thermostat file.

MDEM writes the main output file and the thermostat file to the same directory as the MDEM executable for both the CL and GUI versions. However, using the "Properties" feature of Windows 95/NT to specify the full path, the location of the output files can be changed.

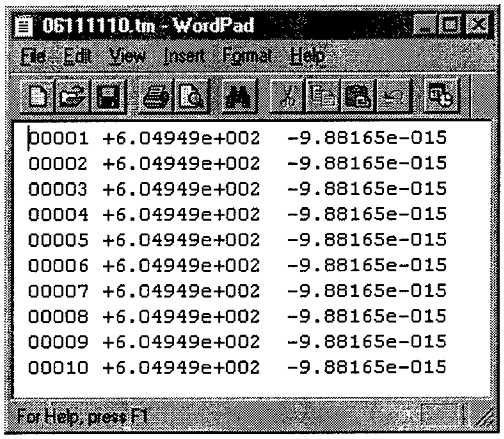
2.2.1 MAIN OUTPUT FILE

The main output is the *.atm file which is written at a user specified frequency (write_data_every). As the output file can be used for an input file the same layout, rules and conventions apply. See Section 2.1 for a detailed discussion and Appendix B for an example.

2.2.2 THERMOSTAT FILE

A secondary output is the thermostat (*.tm) file (Exhibit 6) which is also user specified, both in whether to create and in what frequency to write the thermostat data. The values for the last integration step are also written to file. This file consists of several columns of numbers — the integration step number followed by the system temperature (temp), and the accumulated work done by the thermostats (WTcum[i]) at that integration step for each thermostat. The value for the accumulated work only writes to file if it is not equal to zero (0). The integration step is a unsigned long integer and ke and WTcum are double precision floats.

Exhibit 6. Thermostat File



```
00001 +6.04949e+002  -9.88165e-015
00002 +6.04949e+002  -9.88165e-015
00003 +6.04949e+002  -9.88165e-015
00004 +6.04949e+002  -9.88165e-015
00005 +6.04949e+002  -9.88165e-015
00006 +6.04949e+002  -9.88165e-015
00007 +6.04949e+002  -9.88165e-015
00008 +6.04949e+002  -9.88165e-015
00009 +6.04949e+002  -9.88165e-015
00010 +6.04949e+002  -9.88165e-015
```

2.2.3 SCREEN DISPLAY

Information is displayed to the screen both in the CL version as well as in the GUI version. The CL version writes to screen a summary of the input parameters, number and type of atoms, thermostat and accumulated energy after each integration, and time to execute the integeration. Knowing the expected user responses during the run, the user can pipe the output to a file for viewing at a later time. In this example (Exhibit 7), the user elected to write a thermostat file every integration step.

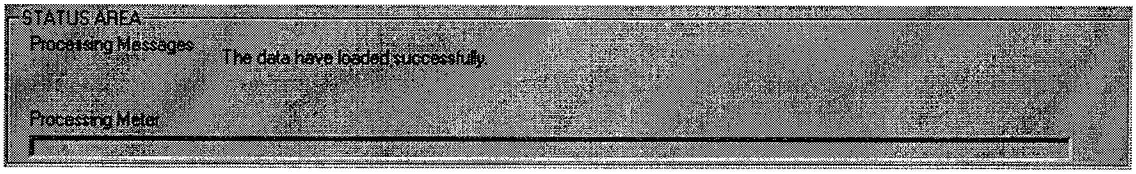
Exhibit 7. Command-line Messages to Screen

```
                         PLEASE REVIEW THE FOLLOWING DATA.

Edit the header of the seed file 13rica.atm if they are incorrect.

There are 13 atoms in the system.
They have evolved for 0.000000E+000 seconds,
originally from the file 13RIC.atm

System ke and temperature are 0.000000E+000 J and 0.000000E+000 kelvins

Recall that Tstat<0 implies xi=constant=xi0.

Thermostat 1 is set for -5.000000E-008 kelvins
and will change at temp_step 1 = 0.000000E+000 kelvins per step.
xi0 1 = 0.000000E+000 N s/m
Thermostat 2 is set for -1.000000E+000 kelvins
and will change at temp_step 2 = 0.000000E+000 kelvins per step.
xi0 2 = 0.000000E+000 N s/m
Thermostat 3 is set for -1.000000E+000 kelvins
and will change at temp_step 3 = 0.000000E+000 kelvins per step.
xi0 3 = 0.000000E+000 N s/m
Thermostat 4 is set for -1.000000E+000 kelvins
and will change at temp_step 4 = 0.000000E+000 kelvins per step.
xi0 4 = 0.000000E+000 N s/m

The Tstat time constant is 1.000000E-014 seconds
10 integration time steps are scheduled
each step of length 1.000000E-015 seconds
A *.atm file will be saved every 2000000 steps.

The force cut off length is 9.000000E-010 meters.
To modify these conditions, quit now and change the seed file.

The interatomic force model is called several times in move_atoms()
by a name that should start with "force_". Both move_atoms() and
force_* are in the file mdemutl.c; look there for details.

-------------------------------------------
```

```
Do you want to create a separate temperature output file?
Enter either a 'Y' or 'N.
Enter the interval to use when writing a *.tm file.
Enter a 1 (one) to update after every step, or
a 5 for every fifth step, etc.

-------------------------------------------
Choose 'continue' if data are correct.
c -continue; q -quit


The atom data have loaded successfully.

There are 13 Ar atoms in this data set.

-------------------------------------------
Choose 'continue' if data are correct.
c -continue; q -quit

00001 +2.134e-013K +5.746e-035J
00002 +8.536e-013K +2.298e-034J
00003 +1.921e-012K +5.171e-034J
00004 +3.414e-012K +9.192e-034J
00005 +5.334e-012K +1.436e-033J
00006 +7.680e-012K +2.068e-033J
00007 +1.045e-011K +2.814e-033J
00008 +1.365e-011K +3.675e-033J
00009 +1.727e-011K +4.650e-033J
00010 +2.132e-011K +5.740e-033J
stop  time: 165724
start time: 165724
Execution time = 0.00 seconds
Mean compute time per 1.000e-015 sec of simulation time
for a system of 13 atoms
was 0.000e+000 sec

mdem done
```

The GUI version of MDEM displays messages to the user in the bottom pane of the interface screen called the “Status Area” (Exhibit 8). These messages range from noting errors to updating the progress of the integration run.

Exhibit 8. GUI Messages to Screen



The data have loaded successfully.

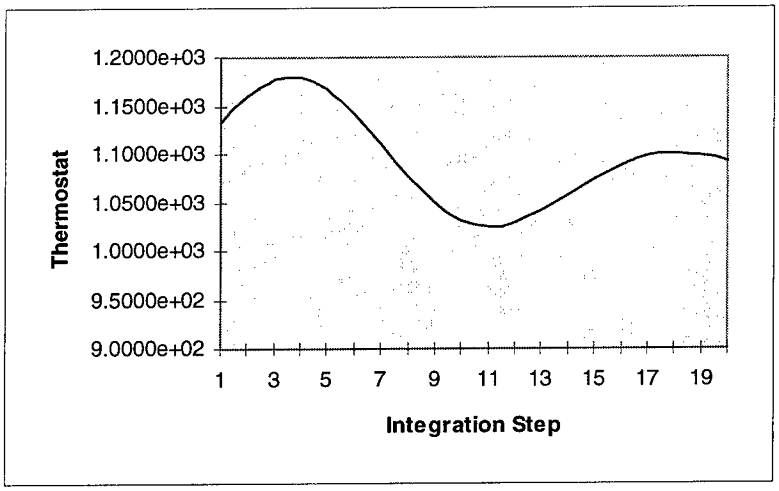
2.2.4 Use of Outputs

One use of computer simulations is to assist the user in the calculation of complex equations. Another use is the visual aspect — one cannot obtain an intuitive notion of either the starting data or the output data from a list of coordinates. One solution, therefore, is to employ a software package which represents the atom coordinates in three dimensions. With a graphic, spatial patterns are more obvious. Hence the use of packages, such as *RasMol* which is available free for download from the Internet.

2.2.4.1 Line Graph

Using the data from the thermostat file, a plot can be graphed. A smooth curve more accurately reflects changes in the data. Exhibit 9 is a plot of the thermostat data from a data set consisting of 9,758 atoms over 20 integrations. The plot was created using Microsoft Excel.

Exhibit 9. Graph of Thermostat Data



2.2.4.2 RasMol Output

RasMol renders the three–dimensional coordinates for a molecule using the pdb file format. This format is not supported by MDEM, although MDEM's molecular coordinates can be stripped from the main output file and reformatted appropriately. Appendix E contains more information on the postprocessing required to move from the *.atm to a *.pdb format.

SECTION THREE
OPERATING INSTRUCTIONS

Section 3 contains the operating instructions for MDEM, from installation, to operation of either version of MDEM, to error resolution.

3.1 INSTALLING MDEM

Installing and executing MDEM depends upon the platform and operating system in use. The directions presented here apply to installing both the executable and the source code. Since the target platform for MDEM is the PC, the instructions here are specific to machines running either Windows™ 95 or Windows™ NT (version 4.0) and will work for either the CL version or GUI version of MDEM.

Basically, there are three methods of moving files in Windows 95.

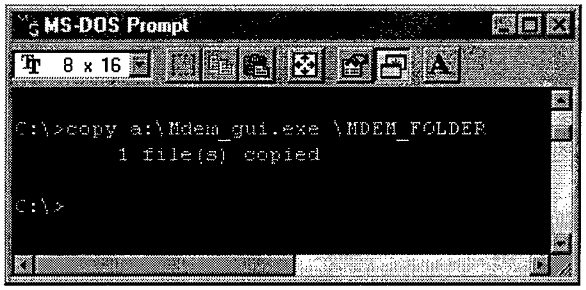
- ❑ (MS–Prompt) Command–line — use the **copy** command to copy a file from one location to another. For example, usie the MS–DOS command–line to copy MDEM_gui.exe from a floppy disk in drive A to a folder called **MDEM_FOLDER**,

    1. type> **copy a:\Mdem_gui.exe \MDEM_FOLDER** <CR>

        ⇒ displays the message: `1 file(s) copied`

        ⇒ see Exhibit 10

Exhibit 10. Copying Executable



```
C:\>copy a:\Mdem_gui.exe \MDEM_FOLDER
        1 file(s) copied

C:\>
```

□ "Drag–and–drop"

1. Using the mouse, click on the target icon.

2. Keeping the mouse button depressed, "drag" the icon to the intended destination.

□ "Copy–paste"

1. Click on the icon with the right mouse button.

   ⇒ displays a popup menu of editing/file management options (Exhibit 11)

2. Move the mouse pointer to the command **`copy`** and release the button.

   ⇒ creates a copy of the file in memory

3. Click the right mouse button on the intended target directory.

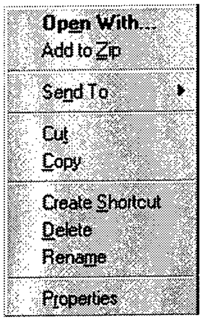
4. Click the right mouse button.

   ⇒ displays the popup menu

5. Select the **`paste`** command.

   ⇒ creates a copy of the file in the selected directory

Exhibit 11. File Management Options



To install MDEM on a workstation, consult the system operator and/or system administration guidelines for the instructions specific to the particular environment.

3.2 USING MDEM

MDEM has two operational modes, CL and GUI. Sections 3.2.1 lists the procedures for using the CL version, while Section 3.2.2 details capabilities and procedures for using the GUI version.

3.2.1 COMMAND–LINE PROCEDURES

The following instructions describe and illustrate the process for executing the CL version of MDEM, by either using the MS–DOS prompt window or using the "drag–and–drop" execution method.

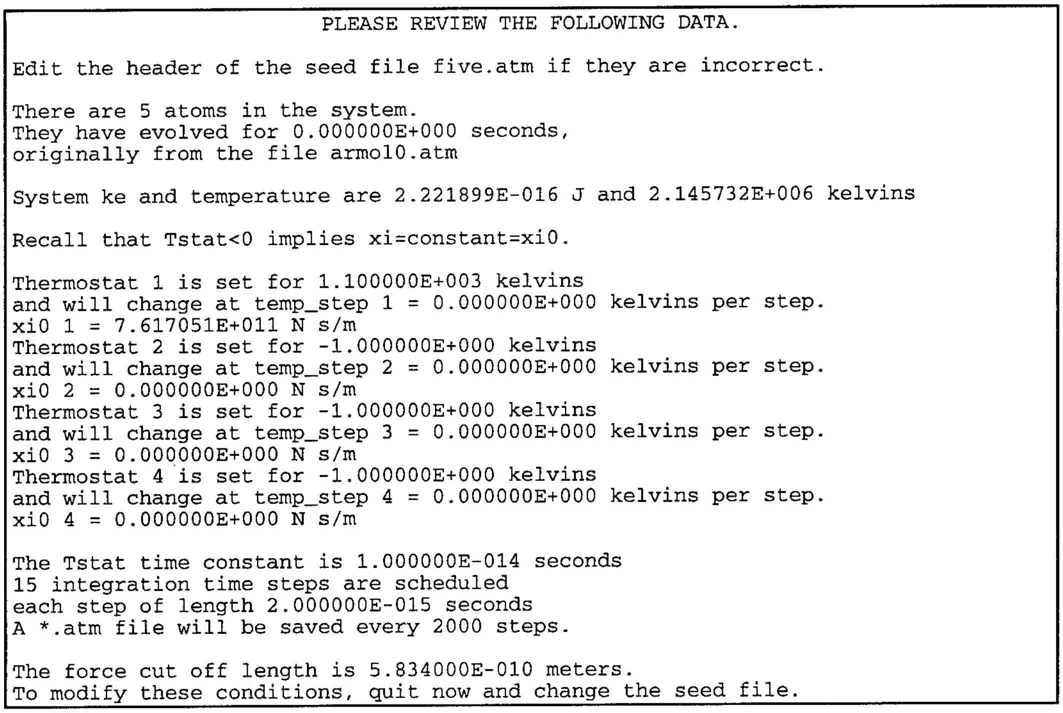
> ☞ If a pinning point file is specified in the main input file, it must be located in the same directory as the MDEM executable.

1. type> **mdem4** *data*

⇒ where **mdem4** is the name of the MDEM executable and *data* is the name of the input data file

♦ Or, use the "drag–and–drop" method to drop the input file onto **mdem4.exe**

⇒ displays a review of the data contained in the header information



```
                      PLEASE REVIEW THE FOLLOWING DATA.

Edit the header of the seed file five.atm if they are incorrect.

There are 5 atoms in the system.
They have evolved for 0.000000E+000 seconds,
originally from the file armol0.atm

System ke and temperature are 2.221899E-016 J and 2.145732E+006 kelvins

Recall that Tstat<0 implies xi=constant=xi0.

Thermostat 1 is set for 1.100000E+003 kelvins
and will change at temp_step 1 = 0.000000E+000 kelvins per step.
xi0 1 = 7.617051E+011 N s/m
Thermostat 2 is set for -1.000000E+000 kelvins
and will change at temp_step 2 = 0.000000E+000 kelvins per step.
xi0 2 = 0.000000E+000 N s/m
Thermostat 3 is set for -1.000000E+000 kelvins
and will change at temp_step 3 = 0.000000E+000 kelvins per step.
xi0 3 = 0.000000E+000 N s/m
Thermostat 4 is set for -1.000000E+000 kelvins
and will change at temp_step 4 = 0.000000E+000 kelvins per step.
xi0 4 = 0.000000E+000 N s/m

The Tstat time constant is 1.000000E-014 seconds
15 integration time steps are scheduled
each step of length 2.000000E-015 seconds
A *.atm file will be saved every 2000 steps.

The force cut off length is 5.834000E-010 meters.
To modify these conditions, quit now and change the seed file.
```

```
The interatomic force model is called several times in move_atoms()
by a name that should start with "force_".  Both move_atoms() and
force_* are in the file mdemutl.c; look there for details.
```

⇒ displays the input parameters to the screen

⇒ displays message allowing the user to create a “separate temperature output file”

```
Do you want to create a separate temperature output file?
Enter either a 'Y' or 'N.
```

2. type> *Y*

⇒ requests that the user enter the number for the write frequency

```
Enter the interval to use when writing a *.tm file.
Enter a 1 (one) to update after every step, or
a 5 for every fifth step, etc.
```

3. type>*1*

⇒ “*1*” writes to file the thermostat information after every step

```
Choose 'continue' if data are correct.
c -continue; q -quit
```

4. type> *c*

⇒ loads the atom data

⇒ displays the “successful load” message

```
The atom data have loaded successfully.
```

⇒ displays the number of atoms by type.

```
There are 5 Ni atoms in this data set.
```

⇒ displays message to continue if input parameters are correct, or to quit if not correct (enter “c” to continue or “q” to quit)

```
Choose 'continue' if data are correct.
c -continue; q -quit
```

5. type> *c*

⇒ starts execution

⇒ writes to file a record of the run

⇒ displays summary of data to screen for each integration step

```
00001 +6.049e+002K +6.264e-020J xi[1]= -8.239e+012 WTcum[1]= -9.882e-015
00002 +6.049e+002K +6.264e-020J xi[1]= -1.724e+013 WTcum[1]= -9.882e-015
00003 +6.049e+002K +6.264e-020J xi[1]= -2.624e+013 WTcum[1]= -9.882e-015
00004 +6.049e+002K +6.264e-020J xi[1]= -3.524e+013 WTcum[1]= -9.882e-015
00005 +6.049e+002K +6.264e-020J xi[1]= -4.424e+013 WTcum[1]= -9.882e-015
00006 +6.049e+002K +6.264e-020J xi[1]= -5.324e+013 WTcum[1]= -9.882e-015
00007 +6.049e+002K +6.264e-020J xi[1]= -6.224e+013 WTcum[1]= -9.882e-015
00008 +6.049e+002K +6.264e-020J xi[1]= -7.125e+013 WTcum[1]= -9.882e-015
00009 +6.049e+002K +6.264e-020J xi[1]= -8.025e+013 WTcum[1]= -9.882e-015
00010 +6.049e+002K +6.264e-020J xi[1]= -8.925e+013 WTcum[1]= -9.882e-015
00011 +6.049e+002K +6.264e-020J xi[1]= -9.825e+013 WTcum[1]= -9.882e-015
00012 +6.049e+002K +6.264e-020J xi[1]= -1.072e+014 WTcum[1]= -9.882e-015
00013 +6.049e+002K +6.264e-020J xi[1]= -1.163e+014 WTcum[1]= -9.882e-015
00014 +6.049e+002K +6.264e-020J xi[1]= -1.253e+014 WTcum[1]= -9.882e-015
00015 +6.049e+002K +6.264e-020J xi[1]= -1.343e+014 WTcum[1]= -9.882e-015
```

⇒ displays the time to manipulate the atoms

```
stop  time: 220106
start time: 220106
Execution time = 0.05 seconds
Mean compute time per 2.000e-015 sec of simulation time
for a system of 5 atoms
was 3.333e-003 sec
```

⇒ displays the message, “mdem done”

```
mdem done
```

> ☞ Using “drag–and–drop” to start MDEM spawns a MS–DOS prompt window which will disappear as soon as the “`mdem done`” message has displayed.

⇒ returns to the command–line prompt

> ☞ To interrupt MDEM at any point during the integration, while holding down <CTRL>, press <C>.

Refer to Section 3.3 for instructions on resolving any errors.

3.2.2 GRAPHICAL USER INTERFACE OPERATION

The GUI version of Mdem contains all of the functional capabilities of the CL version in addition to features that allow the user to interactively correct invalid data and create new data sets with new input parameters.

3.2.2.1 FUNCTIONALITY

The enhanced capabilities of the GUI version are:

- ❑ Main functions
  - ◆ Load Data — loads the data from any *.atm file; uses the Windows' standard Open dialog, allowing the user to traverse the machine's directory.
  - ◆ Save Data — saves to a new file the inputs as seen on the interface along with the original atom data loaded; uses the Window's standard Save As dialog.
  - ◆ Reset Data — restores the original values for the current data set; Once the data has been saved as a new input file, the new input values are the "original."
  - ◆ Run Mdem — processes the integration; if any changes to the original values are detected, the Save As dialog automatically displays. To not save the changes (if already saved, for example), click on the Save As dialog's "Cancel" button.
  - ◆ Cancel Mdem — cancels the current operation; if the integration is in progress, Mdem finishes the current integration step and writes out the last step to the output file.
  - ◆ Exit Mdem — exits the program
  - ◆ Help — displays minimal information on the program
- ❑ View Atoms — displays the number of atoms by type comprising the data set
- ❑ View Thermostat Data 1–7 — displays the values for the seven initial thermostats in a popup window for:
  - ◆ xi0 — initial thermostat values
  - ◆ temp_ref — desired system temperature
  - ◆ temp_step — change in temp_ref per time step
  - ◆ total work — accumulated energy delivered by the thermostats per integration step
  - ◆ temp_stat — signals that thermostat x is active if the value equals one (1)
- ❑ Pinning
  - ◆ Change Pinning File — displays the name of the pinning file associated with the loaded data set, if any; user can use this button to display the **OPEN** dialog, for selecting or changing a pinning file
  - ◆ View Pinning Points — displays the x, y, and z pinning points (coordinates) in a popup window
  - ◆ View Pinning Constants — displays the x, y, and z pinning force constants in a popup window

- ❑ Status Area
  - ◆ Processing Messages — displays processing messages such as, detection of invalid data during the load process, the current integration step number, and the mean compute time per sec of simulation time for a system of n–atoms.
  - ◆ Processing Meter — visual confirmation that the system is still working

The values of all fields on the interface, except the **Execution, System Initial Temperature**, and **Pinning** file name fields, can be changed directly on the interface. This provides the flexibility of running integrations with minor to major variations in the system parameters while saving the changes as new files.

3.2.2.2 Operating Instructions

The following instructions list and illustrate the procedures for using the GUI version of MDEM.

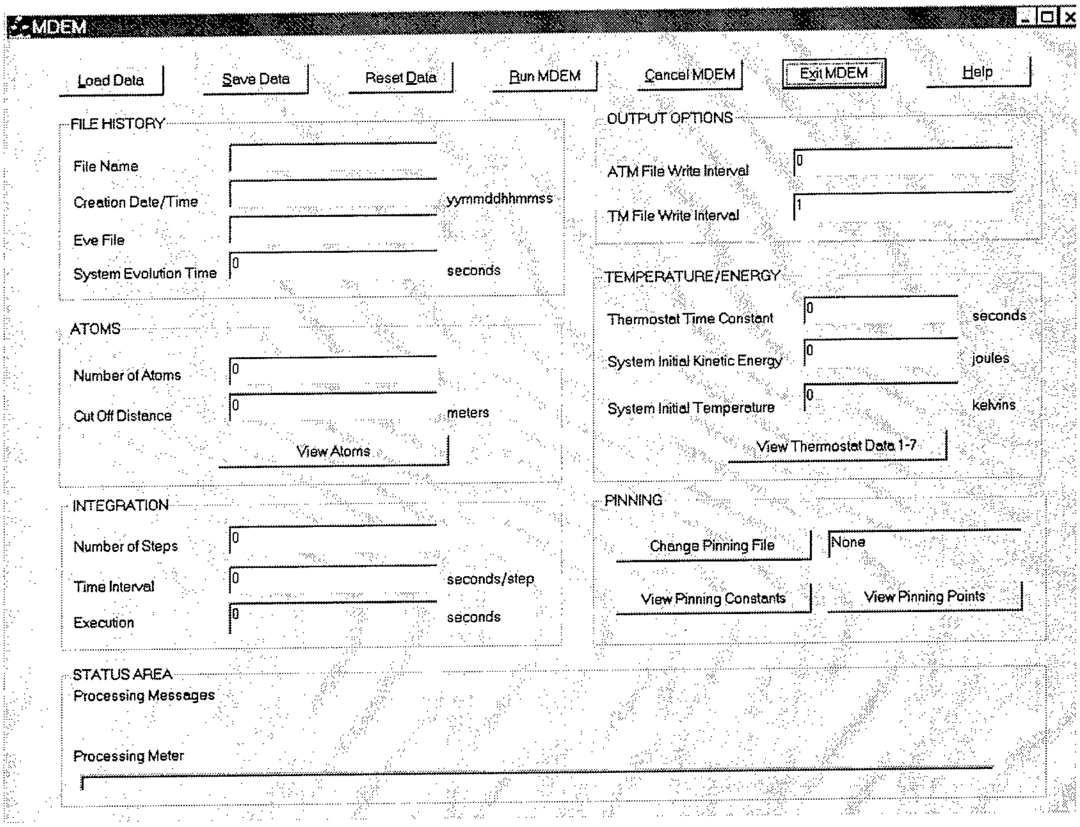
> ☞ If a pinning point file is specified in the main input file, it must be located in the same directory as the MDEM executable. With the GUI version, the pinning point file can be changed via the Change Pinning File button after the initial data load is complete.

1. Start MDEM4_**GUI.EXE** from the (graphical) directory structure, by one of the following methods:

   - ♦ Double–click the left mouse button on MDEM4_**GUI.EXE**
   - ♦ Or, from the MS–DOS command line, type> ***mdem4_gui***

   ⇒ displays the MDEM main dialog (Exhibit 12)

Exhibit 12. MDEM Main Screen



FILE HISTORY
File Name
Creation Date/Time yymmddhhmmss
Eve File
System Evolution Time 0 seconds

ATOMS
Number of Atoms 0
Cut Off Distance 0 meters

INTEGRATION
Number of Steps 0
Time Interval 0 seconds/step
Execution 0 seconds

OUTPUT OPTIONS
ATM File Write Interval 0
TM File Write Interval 1

TEMPERATURE/ENERGY
Thermostat Time Constant 0 seconds
System Initial Kinetic Energy 0 joules
System Initial Temperature 0 kelvins

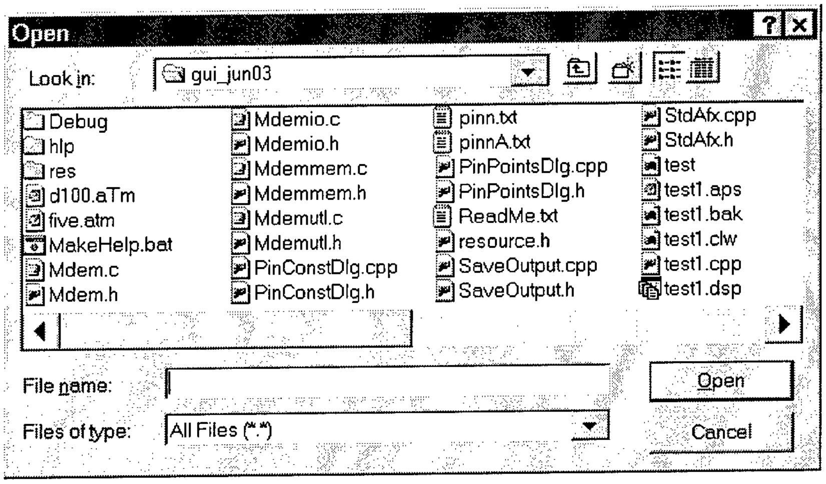
PINNING
None

STATUS AREA
Processing Messages
Processing Meter

2. Select Load Data button

⇒ displays the **Open** dialog (Exhibit 13)

Exhibit 13 Open Dialog Screen



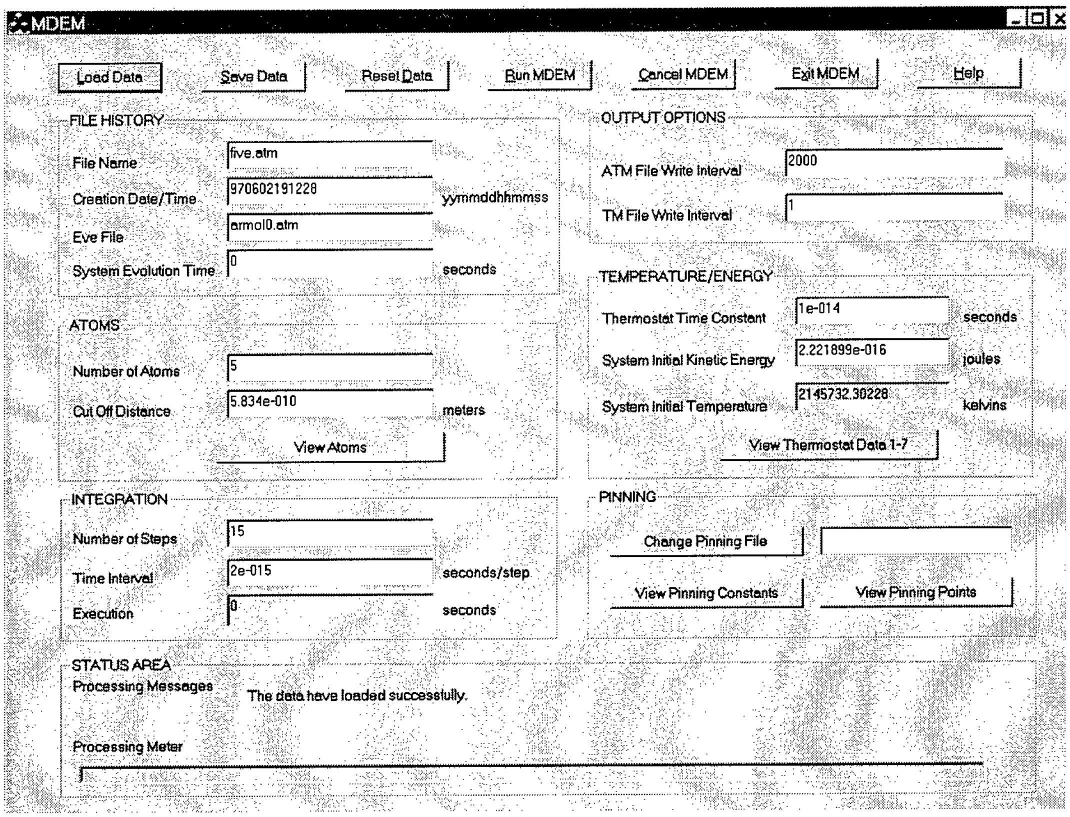
Look in: gui_jun03

Debug
hlp
res
d100.aTm
five.atm
MakeHelp.bat
Mdem.c
Mdem.h
Mdemio.c
Mdemio.h
Mdemmem.c
Mdemmem.h
Mdemutl.c
Mdemutl.h
PinConstDlg.cpp
PinConstDlg.h
pinn.txt
pinnA.txt
PinPointsDlg.cpp
PinPointsDlg.h
ReadMe.txt
resource.h
SaveOutput.cpp
SaveOutput.h
StdAfx.cpp
StdAfx.h
test
test1.aps
test1.bak
test1.clw
test1.cpp
test1.dsp

File name:
Files of type: All Files (*.*)

- ♦ Select the desired input file by either of the following actions
  - Double–click on the file name in the listing, or
  - Type the file name into the **FILE NAME** field on the dialog and then click on the Open button
  - ⇒ retrieves and displays the data in the appropriate fields (Exhibit 14)

Exhibit 14. MDEM Data Display



FILE HISTORY

File Name: five.atm

Creation Date/Time: 970602191228 yymmddhhmmss

Eve File: armol0.atm

System Evolution Time: 0 seconds

ATOMS

Number of Atoms: 5

Cut Off Distance: 5.834e-010 meters

INTEGRATION

Number of Steps: 15

Time Interval: 2e-015 seconds/step

Execution: 0 seconds

OUTPUT OPTIONS

ATM File Write Interval: 2000

TM File Write Interval: 1

TEMPERATURE/ENERGY

Thermostat Time Constant: 1e-014 seconds

System Initial Kinetic Energy: 2.221899e-016 joules

System Initial Temperature: 2145732.30228 kelvins

PINNING

STATUS AREA

Processing Messages: The data have loaded successfully.

Processing Meter

3. Enter a number in the **TM FILE WRITE INTERVAL** field

   ⇒ determines how often to write to file the thermostat information

   ⇒ default equals 1 (one), writing data after each integration step

   ⇒ change to 0 (zero) to not write a separate thermostat file

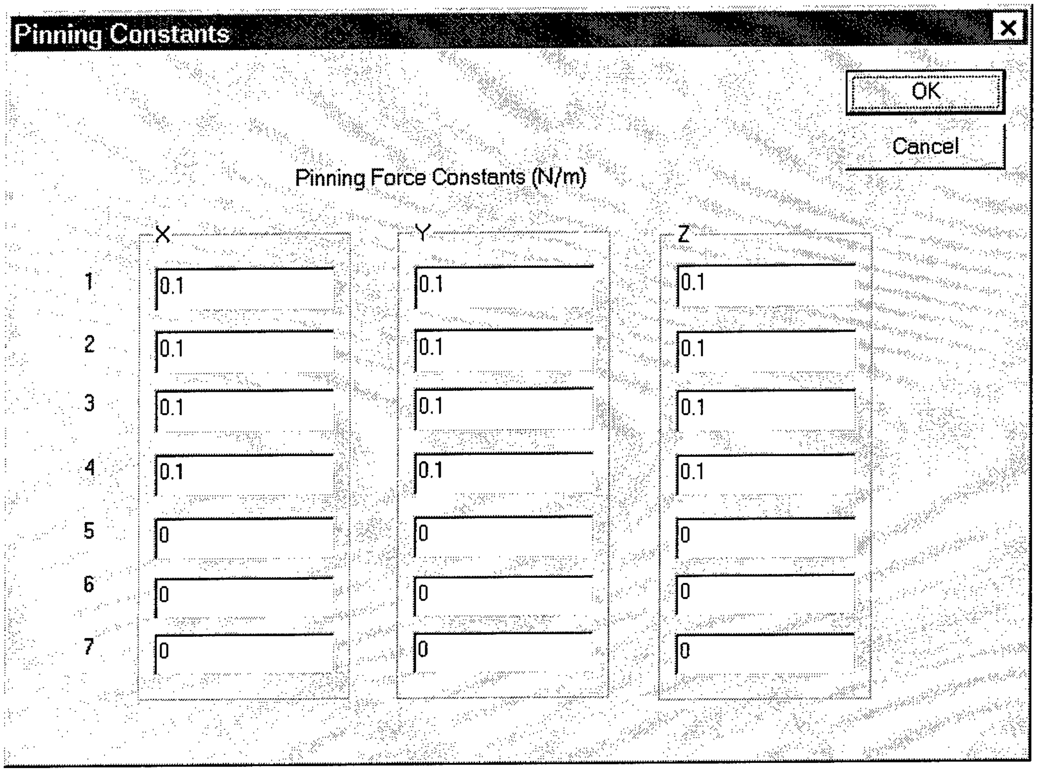
4. Enter changes to the input parameters directly on the interface

   ⇒ **DO NOT** hit "Enter" on the keyboard

5. Select View Pinning Constants button to make changes to pinning points

⇒ displays the **PINNING CONSTANTS** dialog, with values from the input file (Exhibit 15)

Exhibit 15. Pinning Constants



Pinning Force Constants (N/m)

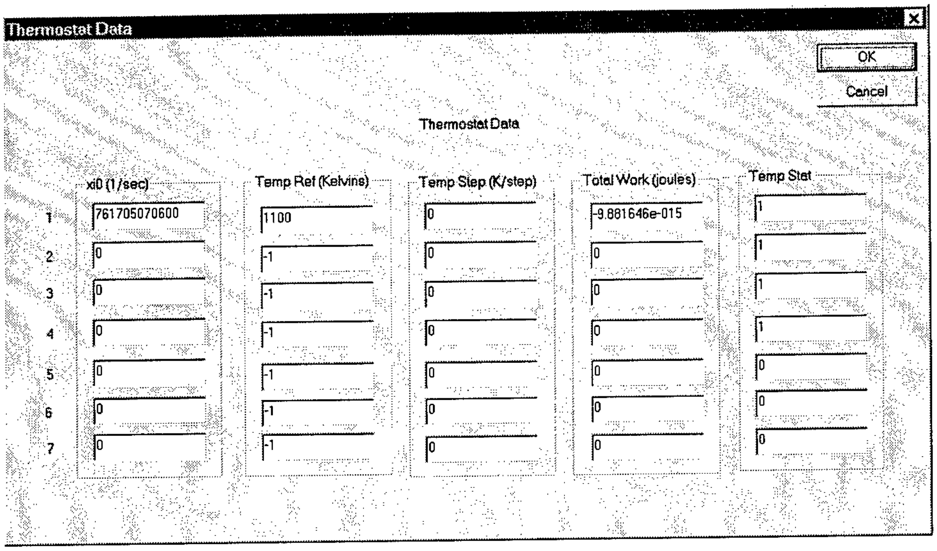
	X	Y	Z
1	0.1	0.1	0.1
2	0.1	0.1	0.1
3	0.1	0.1	0.1
4	0.1	0.1	0.1
5	0	0	0
6	0	0	0
7	0	0	0

- ♦ Click on the desired field to make a change
- ♦ Enter the new value
- ♦ Select the OK button

6. Select View Thermostat Data 1-7 button to make changes to the thermostat data

⇒ displays the **THERMOSTAT DATA** dialog, with values specified in the input file (Exhibit 16)

Exhibit 16. Thermostat Data



	xi0 (1/sec)	Temp Ref (Kelvins)	Temp Step (K/step)	Total Work (joules)	Temp Stat
1	761705070600	1100	0	-9.881646e-015	1
2	0	-1	0	0	1
3	0	-1	0	0	1
4	0	-1	0	0	1
5	0	-1	0	0	0
6	0	-1	0	0	0
7	0	-1	0	0	0

- Click on the desired field to make a change
- Enter the new value
- Select the OK button

  ⇒ removes the dialog and updates the values in the system

7. Select View Pinning Points button to view the pinning points

   ⇒ displays the **PINNING POINTS** dialog (read–only) from the pinning file shown in the pinning file read–only field (Exhibit 17)

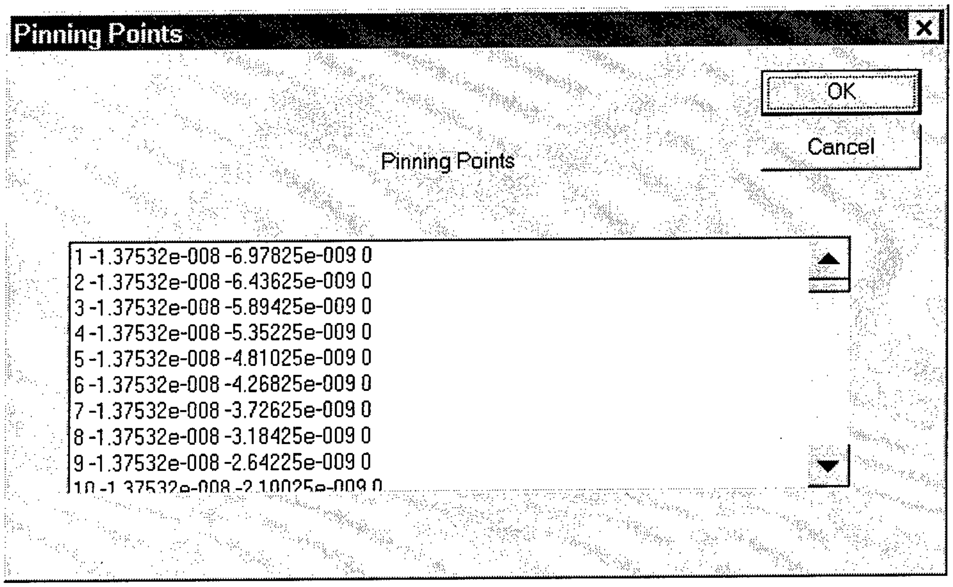
   ⇒ column one: atom number

   ⇒ column two: x coordinate

   ⇒ column three: y coordinate

   ⇒ column four: z coordinate

Exhibit 17. Pinning Points



```
1 -1.37532e-008 -6.97825e-009 0
2 -1.37532e-008 -6.43625e-009 0
3 -1.37532e-008 -5.89425e-009 0
4 -1.37532e-008 -5.35225e-009 0
5 -1.37532e-008 -4.81025e-009 0
6 -1.37532e-008 -4.26825e-009 0
7 -1.37532e-008 -3.72625e-009 0
8 -1.37532e-008 -3.18425e-009 0
9 -1.37532e-008 -2.64225e-009 0
10 -1.37532e-008 -2.10025e-009 0
```

- ♦ Click the Change Pinning File button to select another pinning file

  ⇒ displays the **OPEN** dialog

- ♦ Select desired input file by doing either of the following.

  - double–click on the desired file name in the list, or
  - type the file name into the **FILE NAME** field and then click on the Open button

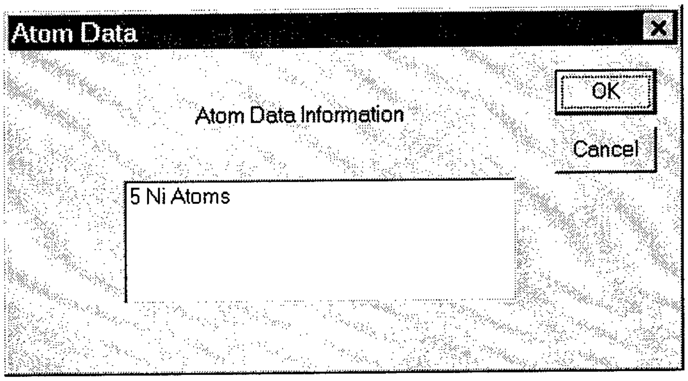
  ⇒ displays the selected file name in the pinning file read–only field

- ♦ Reselect the View Pinning Points button to view the new pinning points

8. Select View Atoms button to view information on the atom types

   ⇒ displays the **ATOM DATA** dialog (read–only) which lists the types of atoms comprising the data set and the number of each type (Exhibit 18)

Exhibit 18. Atom Data



Atom Data Information

5 Ni Atoms

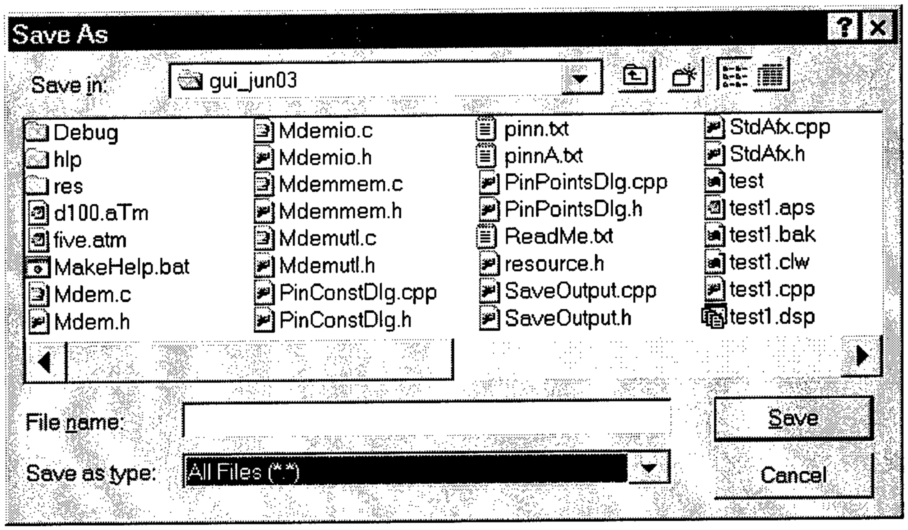
9. Select Save Data button to save any changes for a new input file

⇒ displays the **SAVE AS** dialog (Exhibit 19)

♦ Enter a file name in the **SAVE AS** dialog **FILE NAME** field to save changes

- Select the Save button

Exhibit 19. Save As



Save in: gui_jun03

Debug
hlp
res
d100.aTm
five.atm
MakeHelp.bat
Mdem.c
Mdem.h
Mdemio.c
Mdemio.h
Mdemmem.c
Mdemmem.h
Mdemutl.c
Mdemutl.h
PinConstDlg.cpp
PinConstDlg.h
pinn.txt
pinnA.txt
PinPointsDlg.cpp
PinPointsDlg.h
ReadMe.txt
resource.h
SaveOutput.cpp
SaveOutput.h
StdAfx.cpp
StdAfx.h
test
test1.aps
test1.bak
test1.clw
test1.cpp
test1.dsp

File name:

Save as type: All Files (*.*)

⇒ writes a new file as a record of the input parameters

⇒ can select at any time

⇒ optional

10. Select Run MDEM button

⇒ displays the **SAVE AS** dialog if any changes to the original inputs are detected

♦ Enter a file name in the **SAVE AS** dialog **FILE NAME** field to save changes

  - Select the Save button

♦ Or, select Cancel if changes were saved using Save Data button or to not save the changes

⇒ executes the integration

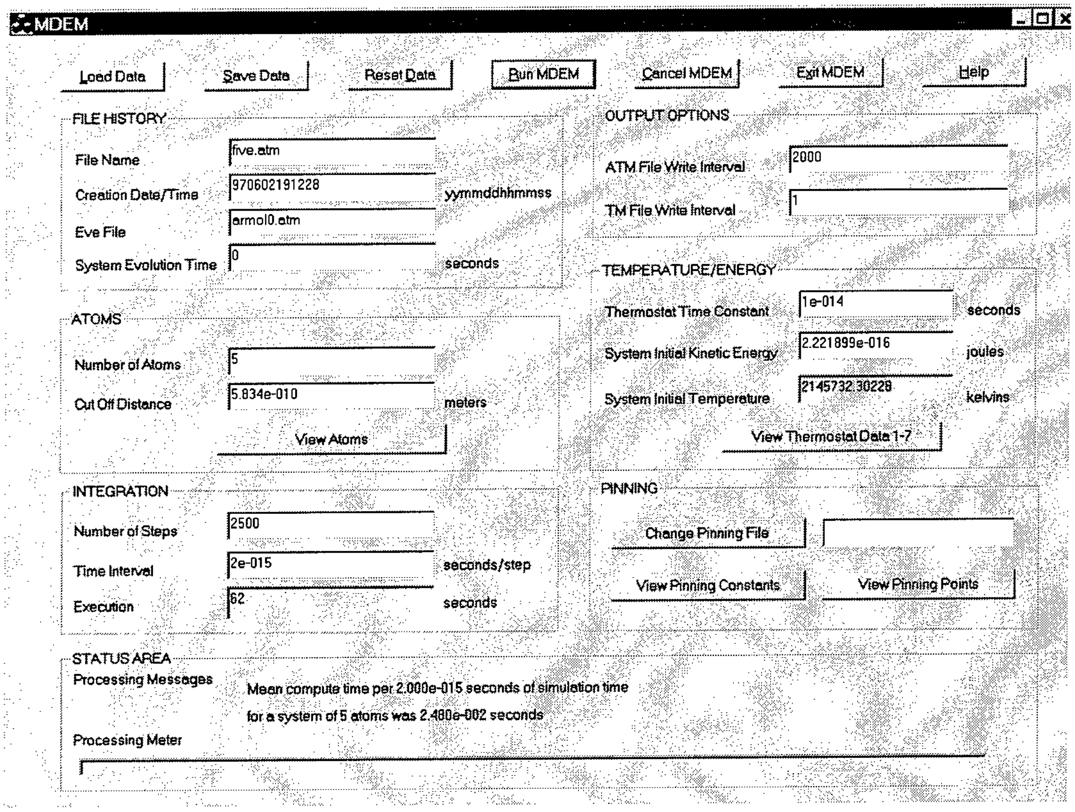
⇒ writes a new file as a record of the input parameters executes the program

⇒ writes the TM file during execution

⇒ writes the output file

⇒ displays the time required to integrate the data set in the **EXECUTION TIME** field (Exhibit 20)

Exhibit 20. MDEM Main Screen Showing Execution Time



FILE HISTORY

File Name: five.atm

Creation Date/Time: 970602191228 yymmddhhmmss

Eve File: armol0.atm

System Evolution Time: 0 seconds

ATOMS

Number of Atoms: 5

Cut Off Distance: 5.834e-010 meters

INTEGRATION

Number of Steps: 2500

Time Interval: 2e-015 seconds/step

Execution: 62 seconds

OUTPUT OPTIONS

ATM File Write Interval: 2000

TM File Write Interval: 1

TEMPERATURE/ENERGY

Thermostat Time Constant: 1e-014 seconds

System Initial Kinetic Energy: 2.221899e-016 joules

System Initial Temperature: 2145732.30228 kelvins

PINNING

STATUS AREA

Processing Messages: Mean compute time per 2.000e-015 seconds of simulation time for a system of 5 atoms was 2.480e-002 seconds

Processing Meter

11. Select the Cancel MDEM button to halt the integration process

⇒ cancels the integration step

⇒ disables all buttons, except Load Data and Exit MDEM

♦ Select Load Data to load another data set

12. Select Exit MDEM button

⇒ exits the program

> ☞ Clicking the X–box in the top right corner of the dialog, on the title bar, kills the program, losing any data not written to file.

3.3 RESOLVING ERRORS

MDEM generates the following error and warning messages when confronted with incorrect or missing data (Table 4).

Table 4. MDEM Error and Warning Messages

MESSAGE	RESOLUTION
Must enter filename on command line; quit & restart	✦ Enter Q to quit MDEM ✦ Use the MS–DOS command line to enter: mdem data, or drag the data icon onto the executable icon
'this_filename' in input file disagrees with command line argument	✦ CL — Quit MDEM ✧ Edit data file's header to match the file names ✦ GUI — Correct the file name in the **FILE NAME** field
write_intrvl must be >= 1	✦ CL — Quit MDEM ✧ Edit data file's header ✦ GUI — Correct the value in the **ATM FILE WRITE INTERVAL** field
del_t must not be 0.0	✦ CL — Quit MDEM ✧ Edit data file's header ✦ GUI — Correct value in the **TIME INTERVAL** field

MESSAGE	RESOLUTION
tau must not be 0.0	✦ t_stat_time_const_tau value cannot equal zero (0) ✦ CL — Quit MDEM ✧ Edit data file's header ✦ GUI — Correct the value in the **THERMOSTAT TIME CONSTANT** field
atom is pinned w/o supporting data	✦ pin_const[t_stat] == 0.0 ✦ CL — Quit MDEM ✧ Edit data file ✧ Restart MDEM ✦ GUI — Edit *.atm file in text editor ✧ Reload data file
atom is pinned but no pinning point is specified	✦ CL — Quit MDEM ✧ Edit pinning point file ✦ GUI — Edit pinning point file using text editor ✧ Save changes ✧ Reload pinning point file via the Change Pinning File button
reached EOF too soon looking for end could not open file	✦ File does not contain the correct number of lines ✦ CL — Quit MDEM ✧ Edit *.atm file in text editor and save changes ✦ GUI — Edit *.atm file in text editor and save changes ✧ Reload data file
desired temperature cannot be zero	✦ temp_ref values cannot equal zero (0) ✦ CL — Quit MDEM ✧ Edit data file's header to change temp_ref values and save changes ✧ Restart MDEM ✦ GUI — Correct value in the **TEMP REF** fields on the **THERMOSTAT** dialog
t_stat_time_const cannot be zero	✦ t_stat_time_const_tau value cannot equal zero (0) ✦ CL — Quit MDEM ✧ Edit data file's header ✦ GUI — Correct the value in the **THERMOSTAT TIME CONSTANT** field

MESSAGE	RESOLUTION
pin_const.(x,y,z) = 0.0; this will have no effect on the system. Unless you change pin_const in the header, pin_const.(x,y,z) will be removed from future *.atm files.	✦ pin_const.(x,y,z) = 0.0 ✦ CL — Quit MDEM ✧ Edit data file's header to change pin_const values and save changes ✧ Restart MDEM ✦ GUI — Correct values in the **PINNING FORCE CONSTANT** fields on the **PINNING CONSTANTS** dialog
reached EOF too soon after dx header	✦ Data file is incomplete ✦ CL — Quit MDEM ✧ Edit *.atm file in text editor and save changes ✦ GUI — Edit *.atm file in text editor and save changes ✧ Reload data file
could not open pinning point file	✦ Specified pinning point file does not exist in the directory being searched

The CL version displays the error message or warning to the screen and then prompts the user to quit, edit the data file, and then restart MDEM.

The GUI version of MDEM flags the same warnings, which display in the Status Area on the MDEM dialog. In addition to errors detected during the loading process, the GUI version automatically flags the entry of wrong data types entered on the MDEM dialog (i.e., user entered a float in a field requiring an integer). These messages are displayed in a popup box which can be dismissed by clicking the dialog's "Ok" button.

SECTION FOUR
FURTHER DEVELOPMENT OF MDEM

The MDEM program provides information on the forces interacting between atoms, thereby providing a vehicle for predicting a substance's static and dynamic properties. In doing so the performance of the MDEM software varies significantly based on compiler, mode, platform, number of atoms, and number of integration steps. Table 5 provides the user with a comparison of the various compilers, their representative compilation/optimization flags, and the experienced execution times for a box about 70 x 106 x 140 angstroms filled with 112423 FCC Nickel atoms. The tags have 2,048 added to them to activate thermostat #1. The atoms were integrated for 20 steps.

A more detailed table can be found in Volume I, Scientific and Final Technical Reports (FINAL).

Table 5. Timing Examples

OPERATIONAL MODE		OPTIMIZATION	TOTAL EXECUTION TIME (SECONDS)
Pentium (75 MZ, 24 MB RAM, Windows 95)			
	CL	–O1	insufficient memory
	CL	–O2	insufficient memory
	MVCC	Blend, Global	1037.82
	GUI	Blend, Global	1169
Pentium Pro (200 MZ, 64 MB RAM, Windows NT)			
	CL	–O1	396.70
	CL	–O2	286.81
	MVCC	Blend, Global	191.64
	GUI	Blend, Global	275

4.1 SOFTWARE INVENTORY

This section contains information on the software files required for further development of the MDEM software as well as the name of the executable required. Table 6 lists the files required for compilation for the two versions. The CL version requires 8 files, while the GUI created using Microsoft Visual C++ (version 5.0) includes the same 8 files plus another 29 files.

Table 6 Software Inventory

FILE	CL	GUI
AfxDlg.rtf		✓
MakeHelp.bat		✓
Mdem.c	✓	✓
Mdemio.c	✓	✓
Mdemmem.c	✓	✓
Mdemutl.c	✓	✓
Mdem.h	✓	✓
Mdemio.h	✓	✓
Mdemmem.h	✓	✓
Mdemutl.h	✓	✓
PinConstDlg.cpp		✓
PinConstDlg.h		✓
PinPointsDlg.cpp		✓
PinPointsDlg.h		✓
resource.h		✓
StdAfx.cpp		✓
StdAfx.h		✓
test1.ico		✓
test1.rc2		✓
test1.cnt		✓
test1.hm		✓
test1.hpj		✓
test1.ph		✓
test1.aps		✓
test1.bak		✓
test1.cpp		✓
test1.dsp		✓
test1.dsw		✓
test1.h		✓
test1.mak		✓
test1.ncb		✓
test1.opt		✓
test1.plg		✓
test1.rc		✓
test1Dlg.cpp		✓
test1Dlg.h		✓
ThermDlg.cpp		✓
ThermDlg.h		✓
ViewAtomsDlg.cpp		✓
ViewAtomsDlg.h		✓

4.1.1 COMMAND–LINE VERSION

To run the CL version of MDEM, the executable MDEM4.exe is required. Consult Table 6 for the exact files required to compile the CL version of MDEM.

4.1.2 GRAPHICAL USER INTERFACE VERSION

The GUI version requires the executable MDEM4_gui.exe. Consult Table 6 for a complete listing of the files required to compile the GUI version of MDEM.

The Microsoft Visual C++ programming environment requires the creation of a project workspace. In version 5.0, this file has the extension of *.dsw. The following is a description of each file created by Microsoft Visual C++, as described in the file "ReadMe.txt":

□ test1.h — This is the main header file for the application. It includes other project specific headers (including Resource.h) and declares the CTest1App application class.

□ test1.cpp — This is the main application source file that contains the application class CTest1App.

□ test1.rc — This is a listing of all of the Microsoft Windows resources that the program uses. It includes the icons, bitmaps, and cursors that are stored in the RES subdirectory. This file can be directly edited in Microsoft Developer Studio.

□ res\test1.ico — This is an icon file, which is used as the application's icon. This icon is included by the main resource file test1.rc.

□ res\test1.rc2 — This file contains resources that are not edited by Microsoft Developer Studio. Place all resources not editable by the resource editor in this file.

□ test1.clw — This file contains information used by ClassWizard to edit existing classes or add new classes. ClassWizard also uses this file to store information needed to create and edit message maps and dialog data maps, and to create prototype member functions.

□ test1Dlg.h, test1Dlg.cpp (the dialog) — These files contain your CTest1Dlg class. This class defines the behavior of the application's main dialog. The dialog's template is in test1.rc, which can be edited in Microsoft Developer Studio.

□ MakeHelp.bat — Use this batch file to create the application's Help file, test1.hLP.

- □ test1.hpj — This file is the Help Project file used by the Help compiler to create the application's Help file.
- □ hlp*.bmp — These are bitmap files required by the standard Help file topics for Microsoft Foundation Class (MFC) Library standard commands.
- □ hlp*.rtf — This file contains the standard help topics for standard MFC commands and screen objects.
- □ StdAfx.h, StdAfx.cpp — These files are used to build a precompiled header (PCH) file named test1.pch and a precompiled types file named StdAfx.obj.
- □ Resource.h — This is the standard header file, which defines new resource IDs. Microsoft Developer Studio reads and updates this file.

These files are required in order to continue development on the current implementation of the GUI version of MDEM.

> ☞ Code written using Microsoft Visual C++ is not necessarily downward compatible.

4.2 ESTABLISHING COMPILER ENVIRONMENT

Setting up the programming environment depends upon the platform and compiler being used. Regardless of using a command–line compiler or a graphical programming environment/compiler, following the compiler's instructions will most efficiently establish the programming environment.

4.2.1 COMMAND–LINE COMPILERS

While the compiler may not require the source code to be located in a specific directory, setting certain environment variables may be necessary. Follow the directions of the compiler being used regarding where to place source code and which environment variables to set.

4.2.2 INTEGRATED DEVELOPMENT ENVIRONMENT

Graphical programming environments, such as Watcom™ C/C++ 10.0, Borland™ C++, and Microsoft™ Visual C++ may have specific requirements for the location of all or certain source files. As with the command–line compilers, certain environment variables may need to be set.

In addition to the source code files, the modern compilers create a "project" or a "workspace," with very specific files required. Use the compiler's instructions in conjunction with the appropriate files, shown in Table 6, to set up the programming environment.

For example, to further develop the GUI version of MDEM, using Microsoft Visual C++ (version 5.0), in the Microsoft Visual C++ default workspace directory, use the following steps.

1. Locate the Microsoft Visual C++ Developer's Studio.

   ⇒ `C:\>Program Files\DevStudio\MyProjects`

2. Create a working directory **`Mdem`**

3. Create the subdirectory **`hlp`**

4. Place the following files into:

   `Program Files\DevStudio\MyProjects\Mdem\hlp`

   - AfxDlg.rtf
   - test1.cnt
   - test1.hm
   - test1.hpj
   - test1.ph

5. Create the subdirectory **`res`**

6. Place the following files into:

   `Program Files\DevStudio\MyProjects\Mdem\res`

   - test1.ico
   - test1.rc2

Place all other source files into the working directory **`Mdem`**

APPENDIX A
ACRONYMS

DJGPP	D.J.'s Gnu C++ (compiler)
FAQ	Freqently Asked Questions
IUPAC	International Union of Pure and Applied Chemistry
MFC	Microsoft Foundation Class
MVCC	Microsoft Visual C++

APPENDIX B
SAMPLE FILES

B1. INPUT/OUTPUT FILE

Exhibit B1 is a 13–atom data set consisting of a regular icosahedron tagged and scaled for Argon. The velocities (current_temperature and ke) are zero. The coordinates are ideal and correspond to minimal potential energy.

Exhibit B1. 13–Atom Input File

```
points =  13
format = ascii
header = marker "Start data\n"
field = locations, velocities, species
structure = 3-vector, 3-vector, scalar
type = float, float, int
dependency = positions, positions, positions
interleaving = field
end

This is a 13-atom regular icosahedron tagged and scaled for Argon.
The velocities (temperature and ke) are zero.
The coordinates are ideal and correspond to minimal PE.


 - Status Data -

date_time 960820

this_filename 13RIC.atm
last_filename 13RIC.atm
frst_filename 13RIC.atm
t=0__filename 13RIC.atm

#_of_atoms  13
evolution_time 0.000000000
cut_off 9.00E-10

kin_enrgy 0.00000000E+00
current_temperature 0.00000000E+00
t_stat_work 1 0.00000000E+00
t_stat_work 2 0.00000000E+00
t_stat_work 3 0.00000000E+00
t_stat_work 4 0.00000000E+00
```

```
- Control Data -

integration_steps_this_run 10
write_data_every 2000000

time_step 1.0000E-15

value_of_xi 1 0.00000000E+00
desired_temperature 1 -5.000000E-08
temp_step 1 0.00000000E+00
pin_const 1 .1 .1 .1

value_of_xi 2 0.00000000000E+00
desired_temperature 2 -1.000000E+00
temp_step 2 0.00000000E+00
pin_const 2 .1 .1 .1

value_of_xi 3 0.00000000000E+00
desired_temperature 3 -1.000000E+00
temp_step 3 0.00000000E+00
pin_const 3 .1 .1 .1

value_of_xi 4 0.00000000000E+00
desired_temperature 4 -1.000000E+00
temp_step 4 0.00000000E+00
pin_const 4 .1 .1 .1

pin_pt_filename
t-stat_time_const_tau 1.0000E-14



Start data
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
 3.2947775E-10  0.0000000E+00  1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
 1.0181461E-10  3.1335182E-10  1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
-2.6655297E-10  1.9366227E-10  1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
-2.6655297E-10 -1.9366227E-10  1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
 1.0181461E-10 -3.1335182E-10  1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
 2.6655297E-10  1.9366227E-10 -1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
-1.0181461E-10  3.1335182E-10 -1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
-3.2947775E-10  0.0000000E+00 -1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
-1.0181461E-10 -3.1335182E-10 -1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
 2.6655297E-10 -1.9366227E-10 -1.6473937E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
 0.0000000E+00  0.0000000E+00  3.6836759E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
 0.0000000E+00  0.0000000E+00 -3.6836759E-10  0.0000000E+00  0.0000000E+00  0.0000000E+00    2066
```

B2. TM FILE

The TM file listing in Exhibit B2 was produced from the sample input file shown in Exhibit B1. The data set contains 13 atoms and had 10 integration steps. The accumulated work column is blank as the values equaled 0.

Exhibit B2. TM File Listing

```
00001 +2.13411e-013
00002 +8.53616e-013
00003 +1.92053e-012
00004 +3.41403e-012
00005 +5.33390e-012
00006 +7.67992e-012
00007 +1.04518e-011
00008 +1.36491e-011
00009 +1.72716e-011
00010 +2.13186e-011
```

The TM file listing in Exhibit B3 was produced from a data set of 9,758 atoms. Exhibit B4 contains the header data for this sample.

Exhibit B3. TM File from 9,758 Atoms

```
00001 +1.13225e+003  -9.88165e-015
00002 +1.15961e+003  -9.88164e-015
00003 +1.17657e+003  -9.88164e-015
00004 +1.17903e+003  -9.88162e-015
00005 +1.16635e+003  -9.88160e-015
00006 +1.14131e+003  -9.88156e-015
00007 +1.10924e+003  -9.88151e-015
00008 +1.07651e+003  -9.88147e-015
00009 +1.04903e+003  -9.88142e-015
00010 +1.03097e+003  -9.88139e-015
00011 +1.02417e+003  -9.88138e-015
00012 +1.02804e+003  -9.88137e-015
00013 +1.04009e+003  -9.88136e-015
00014 +1.05666e+003  -9.88136e-015
00015 +1.07381e+003  -9.88136e-015
00016 +1.08816e+003  -9.88136e-015
00017 +1.09744e+003  -9.88136e-015
00018 +1.10082e+003  -9.88136e-015
00019 +1.09876e+003  -9.88136e-015
00020 +1.09264e+003  -9.88136e-015
```

Exhibit B4. Header Data for 9,758 Atom Data Set

```
points = 9758
format = ascii
header = marker "Start data\n"
field = locations, velocities, species
structure = 3-vector, 3-vector, scalar
type = float, float, int
dependency = positions, positions, positions
interleaving = field
end
```

```
 - Status Data -

date_time 961107014110

this_filename 11070141.atm
last_filename 10291002.atm
frst_filename 10291002.atm
t=0__filename armol0.atm

#_of_atoms 9758
evolution_time 1.430000004E-10
cut_off 5.834000E-10

kin_enrgy 2.221899E-16
current_temperature 1.099473E+03
t_stat_work 1 -9.881646E-15
t_stat_work 2 0.000000E+00
t_stat_work 3 0.000000E+00
t_stat_work 4 0.000000E+00



- Control Data -

integration_steps_this_run 20
write_data_every 2000

time_step 2.0000E-15

value_of_xi 1 7.617050706E+11
desired_temperature 1 1.100000E+03
temp_step 1 0.000000E+00
pin_const 1 .1 .1 .1

value_of_xi 2 0.000000000E+00
desired_temperature 2 -1.000000E+00
temp_step 2 0.000000E+00
pin_const 2 .1 .1 .1

value_of_xi 3 0.000000000E+00
desired_temperature 3 -1.000000E+00
temp_step 3 0.000000E+00
pin_const 3 .1 .1 .1

value_of_xi 4 0.000000000E+00
desired_temperature 4 -1.000000E+00
temp_step 4 0.000000E+00
pin_const 4 .1 .1 .1

pin_pt_filename
t-stat_time_const_tau 1.0000E-14
```

B3. PINNING POINT FILE

Exhibit B5. Pinning File

```
  1 -1.375325E-08 -6.978250E-09  0.000000E+00
  2 -1.375325E-08 -6.436250E-09  0.000000E+00
  3 -1.375325E-08 -5.894250E-09  0.000000E+00
  4 -1.375325E-08 -5.352250E-09  0.000000E+00
  5 -1.375325E-08 -4.810250E-09  0.000000E+00
  6 -1.375325E-08 -4.268250E-09  0.000000E+00
  7 -1.375325E-08 -3.726250E-09  0.000000E+00
  8 -1.375325E-08 -3.184250E-09  0.000000E+00
  9 -1.375325E-08 -2.642250E-09  0.000000E+00
 10 -1.375325E-08 -2.100250E-09  0.000000E+00
 11 -1.375325E-08 -1.558250E-09  0.000000E+00
 12 -1.375325E-08 -1.016250E-09  0.000000E+00
 13 -1.375325E-08 -4.742500E-10  0.000000E+00
 14 -1.375325E-08  6.774978E-11  0.000000E+00
 15 -1.375325E-08  6.097496E-10  0.000000E+00
 16 -1.375325E-08  1.151750E-09  0.000000E+00
 17 -1.375325E-08  1.693750E-09  0.000000E+00
 18 -1.375325E-08  2.235750E-09  0.000000E+00
 19 -1.375325E-08  2.777750E-09  0.000000E+00
 20 -1.375325E-08  3.319750E-09  0.000000E+00
 21 -1.375325E-08  3.861750E-09  0.000000E+00
 22 -1.375325E-08  4.403750E-09  0.000000E+00
 23 -1.375325E-08  4.945750E-09  0.000000E+00
 24 -1.375325E-08  5.487750E-09  0.000000E+00
 25 -1.375325E-08  6.029750E-09  0.000000E+00
```

B4. OUTPUT FILE

The following output file was created from the sample input file of 13 atoms.

Exhibit B6. 13–Atom Output File

```
points = 13
format = ascii
header = marker "Start data\n"
field = locations, velocities, species
structure = 3-vector, 3-vector, scalar
type = float, float, int
dependency = positions, positions, positions
interleaving = field
end



 - Status Data -
```

```
date_time 970615165724

this_filename 06151657.24
last_filename 13RICA.atm
frst_filename 13rica.atm
t=0__filename 13RIC.atm

#_of_atoms 13
evolution_time 1.000000000E-014
cut_off 9.000000E-010

kin_enrgy 5.739603E-033
current_temperature 2.131865E-011
t_stat_work 1 0.000000E+000
t_stat_work 2 0.000000E+000
t_stat_work 3 0.000000E+000
t_stat_work 4 0.000000E+000



- Control Data -

integration_steps_this_run 10
write_data_every 2000000

time_step 1.0000E-015

value_of_xi 1 0.000000000E+000
desired_temperature 1 -5.000000E-008
temp_step 1 0.000000E+000
pin_const 1 1.000000E-001 1.000000E-001 1.000000E-001

value_of_xi 2 0.000000000E+000
desired_temperature 2 -1.000000E+000
temp_step 2 0.000000E+000
pin_const 2 1.000000E-001 1.000000E-001 1.000000E-001

value_of_xi 3 0.000000000E+000
desired_temperature 3 -1.000000E+000
temp_step 3 0.000000E+000
pin_const 3 1.000000E-001 1.000000E-001 1.000000E-001

value_of_xi 4 0.000000000E+000
desired_temperature 4 -1.000000E+000
temp_step 4 0.000000E+000
pin_const 4 1.000000E-001 1.000000E-001 1.000000E-001

pin_pt_filename
t-stat_time_const_tau 1.0000E-014
```

```
Start data
 1.4612984E-031  2.4355249E-031 -1.1690387E-030  4.8709947E-017  8.6241014E-022 -7.3064920E-017 2066
 3.2947775E-010 -5.4801443E-032  1.6473937E-010  6.7466970E-005 -1.2178349E-017 -1.2172221E-004 2066
 1.0181461E-010  3.1335182E-010  1.6473937E-010 -5.2646169E-005  6.3185794E-005 -1.1828024E-004 2066
-2.6655297E-010  1.9366227E-010  1.6473937E-010 -3.2905193E-005 -1.6241691E-006 -9.1760050E-005 2066
-2.6655297E-010 -1.9366227E-010  1.6473937E-010 -3.2905193E-005  1.6241691E-006 -9.1760050E-005 2066
 1.0181461E-010 -3.1335182E-010  1.6473937E-010 -5.2646169E-005 -6.3185794E-005 -1.1828024E-004 2066
 2.6655297E-010  1.9366227E-010 -1.6473937E-010  3.2905193E-005 -1.6241691E-006  9.1760050E-005 2066
-1.0181461E-010  3.1335182E-010 -1.6473937E-010  5.2646169E-005  6.3185794E-005  1.1828024E-004 2066
-3.2947775E-010  1.8263888E-032 -1.6473937E-010 -6.7466970E-005 -7.1239549E-022  1.2172221E-004 2066
-1.0181461E-010 -3.1335182E-010 -1.6473937E-010  5.2646169E-005 -6.3185794E-005  1.1828024E-004 2066
 2.6655297E-010 -1.9366227E-010 -1.6473937E-010  3.2905193E-005  1.6241691E-006  9.1760050E-005 2066
 2.6866806E-020 -4.8712769E-032  3.6836759E-010  4.8881984E-006 -8.8211700E-022  8.1322608E-005 2066
-2.6866806E-020 -1.9791082E-032 -3.6836759E-010 -4.8881984E-006  3.0435455E-018 -8.1322608E-005 2066
```

B5. COMMAND–LINE SCREEN DISPLAY

Exhibit B7 is the command–line screen output for the thirteen atom input file. The run was piped to file, hence only the messages displayed to the screen by MDEM appear. In this run, the user elected to create the TM file, with a write frequency of one.

Exhibit B7. Command–Line Screen Display

```
                              PLEASE REVIEW THE FOLLOWING DATA.

Edit the header of the seed file 13rica.atm if they are incorrect.

There are 13 atoms in the system.
They have evolved for 0.000000E+000 seconds,
originally from the file 13RIC.atm

System ke and temperature are 0.000000E+000 J and 0.000000E+000 kelvins

Recall that Tstat<0 implies xi=constant=xi0.

Thermostat 1 is set for -5.000000E-008 kelvins
and will change at temp_step 1 = 0.000000E+000 kelvins per step.
xi0 1 = 0.000000E+000 N s/m
Thermostat 2 is set for -1.000000E+000 kelvins
and will change at temp_step 2 = 0.000000E+000 kelvins per step.
xi0 2 = 0.000000E+000 N s/m
Thermostat 3 is set for -1.000000E+000 kelvins
and will change at temp_step 3 = 0.000000E+000 kelvins per step.
xi0 3 = 0.000000E+000 N s/m
Thermostat 4 is set for -1.000000E+000 kelvins
and will change at temp_step 4 = 0.000000E+000 kelvins per step.
xi0 4 = 0.000000E+000 N s/m

The Tstat time constant is 1.000000E-014 seconds
10 integration time steps are scheduled
```

```
each step of length 1.000000E-015 seconds
A *.atm file will be saved every 2000000 steps.

The force cut off length is 9.000000E-010 meters.
To modify these conditions, quit now and change the seed file.

The interatomic force model is called several times in move_atoms()
by a name that should start with "force_". Both move_atoms() and
force_* are in the file mdemutl.c; look there for details.

-------------------------------------------
Do you want to create a separate temperature output file?
Enter either a 'Y' or 'N.

Enter the interval to use when writing a *.tm file.
Enter a 1 (one) to update after every step, or
a 5 for every fifth step, etc.

-------------------------------------------
Choose 'continue' if data are correct.
c -continue; q -quit


The atom data have loaded successfully.

There are 13 Ar atoms in this data set.

-------------------------------------------
Choose 'continue' if data are correct.
c -continue; q -quit

00001 +2.134e-013K +5.746e-035J
00002 +8.536e-013K +2.298e-034J
00003 +1.921e-012K +5.171e-034J
00004 +3.414e-012K +9.192e-034J
00005 +5.334e-012K +1.436e-033J
00006 +7.680e-012K +2.068e-033J
00007 +1.045e-011K +2.814e-033J
00008 +1.365e-011K +3.675e-033J
00009 +1.727e-011K +4.650e-033J
00010 +2.132e-011K +5.740e-033J
stop  time: 165724
start time: 165724
Execution time = 0.00 seconds
Mean compute time per 1.000e-015 sec of simulation time
for a system of 13 atoms
was 0.000e+000 sec

mdem done
```

APPENDIX C
DJGPP

C1. GENERAL INFORMATION

DJGPP is a freeware port of GNU C to the 32–bit development system. It is designed for the Intel 30836 PCs (and higher) running the DOS architecture. Many of the GNU development utilities have been ported to DJGPP.

A fairly extensive list of sites from which to download DJGPP exists. Delorie Software's Home Page (www.delorie.com) is the best starting point for more information on DJGPP.

C2. INSTALLATION

To get started with DJGPP, download the binary distributions. To build C programs, the following zip files are required.

- ❑ README.1ST
- ❑ djdev201.zip
- ❑ gcc2721b.zip
- ❑ bnu27b.zip

The README.1ST file details the procedures for installing DJGPP. An important note is to understand the amount of DMPI memory configured for the target machine as compared to that required by DJGPP. The FAQ archives (www.delorie.com/djgpp/v2faq) contain complete information on the minimal and optimal configurations for developing and running programs using DJGPP.

C3. COMPILE FLAGS USED

In the development of MDEM, the major goal was to decrease the operation time. One method used to address this objective was the use of the various compiler flags.

❑ compile flag –O1 — reduces code size and execution time of the program

❑ compile flag –O2 — signals all optional optimization, except loop unrolling and function inlining

❑ compile flag –O3 — includes all optimizations from -O2 as well as the "inline-function" option

Binaries for optimization for the Pentium became available at the end of the MDEM development effort.

APPENDIX D
MICROSOFT VISUAL C++

D1. GENERAL INFORMATION

Microsoft Visual C++ (version 5.0) was the development environment for the GUI version of MDEM. With Visual C++, the developer can choose from the following.

Table D1. Configuration Choices

CATEGORY	TYPE	OPTION
C++ Language		
	Pointer-to-Member Representation	✦ Best Case Always ✦ General Purpose Always
	Enable exception handling	
Code Generation		
	Processor	✦ 80386 ✦ 80486 ✦ Pentium ✦ Pentium Pro ✦ Blend
	Run-time Library	✦ Single-threaded ✦ Multithreaded ✦ Multithreaded DLL
	Calling Convention	✦ _cdecl ✦ _fastcall ✦ _stdcall
	Struct Member Alignment	✦ 1 byte ✦ 2 bytes ✦ 4 bytes ✦ 8 bytes ✦ 16 bytes

CATEGORY	TYPE	OPTION
Optimizations		
	Maximize Speed	✦ Frame-Pointer Omission
	Minimize Code	none
	Customize	✦ Assume No Aliasing ✦ Assume Aliasing Across Function ✦ Global ✦ Generate Intrinsic Functions ✦ Improve Float Consistency ✦ Favor Fast Code ✦ Favor Small Code ✦ Frame-Pointer Omission ✦ Full Optimization

Visual C++ has a processor code generation options option for the each of the Intel PC processors. However, the flag for the Intel 80386 processor is being phased out. The following describes the processor choices considered during Mdem development: ("/G (Optimize for Processor) Options", Microsoft Visual C++'s (version 5.0) help page).

- □ Blend — "Optimizes the code created to favor the Pentium. It blends the optimizations for the 80386, 80486, Pentium, and Pentium Pro options. This option forces a value of 500 for the _M_IX86 preprocessor macro."

- □ Pentium — "Optimizes the code created to favor the Pentium. Use this option for programs meant only for the Pentium. This option forces a value of 500 for the _M_IX86 preprocessor macro."

- □ Pentium Pro — "Optimizes the code created to favor the Pentium Pro. Use this option for programs meant only for the Pentium Pro. This option forces a value of 600 for the _M_IX86 preprocessor macro."

No hard and fast statement can be given regarding which combination of processor/ optimization options produced the fastest executable. In general though, the processor option "Blend" and the optimization "Global" produced the fastest code for both the Pentium and the Pentium Pro. MDEM.exe and MDEM_GUI.exe were delivered using these options.

APPENDIX E
POSTPROCESSING VIEWER

RasMol is a molecular-graphics viewer available free from the Internet. The program renders the three-dimensional coordinates for a molecule using the pdb file format. It displays the molecule in various representations and allows the user to rotate the molecule interactively.

Two sites from which *RasMol* can be obtained are:

- ❑ University of Edinburgh (ftp.dcs.ed.ac.uk)
- ❑ Sunsite Europe (sunsite.doc.ic.ac.uk)

MDEM output files must be reformatted to the pdb file format, which involves creating a new file from the *.atm file. The pdb file consists of records of 80 characters each. The following table shows the structure of the record.

Table E1. Pdb File Format

COLUMN	FIELD
1 — 6	keyword: '**ATOM**' or '**HETATM**'
7 — 11	atom serial number
13 — 16	atom name, in the IUPAC format
17	alternate location indicator indicated by A, B, or C
18 — 20	residue name, in IUPAC format
23 — 26	residue sequence number
27	code for insertions of residues
31 - 38	X-coordinate
39 — 46	Y-coordinate
47 — 54	Z-coordinate
55 — 60	occupancy
61 — 66	Temperature Factor
68 — 70	Footnote Number

More detailed information can be found in *RasMol* v2.5 User's Manual. Exhibit E1 is a sample pdb input file.

Exhibit E1. Sample PDB Input File

```
HETATM    1  Ni          1       -3.096 -26.745   4.008  1.00 -0.26
HETATM    2  Ni          1       -5.356  12.247   7.768  1.00 -0.49
HETATM    3  Ni          1       -3.754  15.101  -3.710  1.00 -0.33
HETATM    4  Ni          1       -5.533  30.396 -10.181  1.00 -0.50
HETATM    5  Ni          1       -1.560  24.986  -8.356  1.00 -0.11
HETATM    6  Ni          1       -3.530  -3.735  -6.815  1.00 -0.30
HETATM    7  Ni          1       -4.241   3.171  -0.581  1.00 -0.37
HETATM    8  Ni          1       -3.545  29.916   3.347  1.00 -0.30
HETATM    9  Ni          1       -0.901 -37.974  -9.365  1.00 -0.04
HETATM   10  Ni          1        3.740 -38.533  -9.719  1.00  0.42
HETATM   11  Ni          1       -4.370   1.853 -14.719  1.00 -0.39
HETATM   12  Ni          1       -1.985  29.005  -6.385  1.00 -0.15
HETATM   13  Ni          1       -5.229  44.808   4.216  1.00 -0.47
```

☆U.S. GOVERNMENT PRINTING OFFICE: 1998-610-130-61143

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.